A REVISTA DOS USUÁRIOS DO TK

MICROHOBBY

ANO I - N.3 - 1983

"LIFE" EM LINGUAGEM DE MÁQUINA

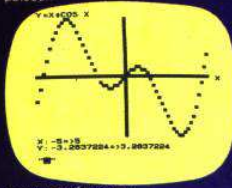
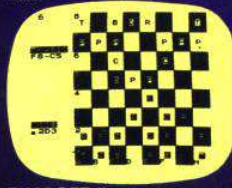
Quebra Cabeça: O Motoqueiro no Deserto

Fita do Mês: O Simulador de Vôo

Aritmética Financeira

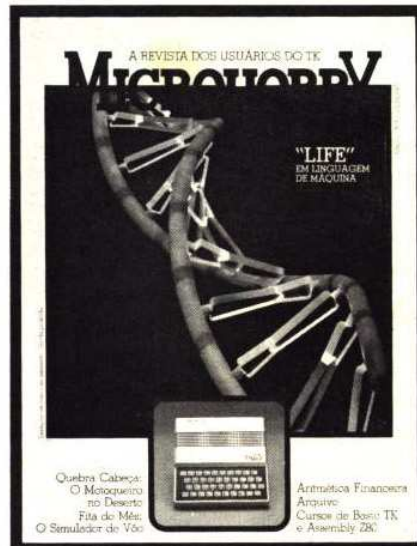
Arquivo

Cursos de Basic TK e Assembly Z80

MICROSOFT
Programas para o seu
TK82-C e TK85
JOGO DE GAMÃO
16K
Este programa apresenta o tabuleiro no video e utiliza o eficiente codigo de maquina, permitindo 4 (quatro) niveis de dificuldades de jogo.
MONSTRO DAS TREVAS
TRIDIMENSIONAL · 16K
Impressionante jogo onde você deve evitar o monstro das trevas. Tudo em 3 dimensões.
DEMOLIDOR
2K
Jogo animado, tipo "fliperama". O jogador devera demolir uma parede com uma bola que se encontra sempre em movimento.
LABIRINTO
TRIDIMENSIONAL · 16K
Jogo em três dimensões. O jogador podera definir a dificuldade do Labirinto. O programa apresenta a posição do jogador em perspectiva. Em qualquer momento e possivel pedir auxilio ao computador.
INVASORES DO
ESPAÇO · 16K
Consiste de uma frota de naves invasoras extraterrenas, descendo no planeta Terra. Sua missão e destruir as naves invasoras dispondo da arma de raios laser.
RALLY
16K
Emocionante corrida de rally em um labirinto, onde poderão ser testados sua habilidade e seus reflexos. Para conseguir seu intento, você deverá evitar carros-ataque e obstaculos em seu trajeto.
MATEMÁTICA I
16K/64K
Análise grafica de funções matematicas, resolução de sistemas de equações lineares (16K-51 equações/64K-95 equações), e Calculo de integrais definidos.
TK-MAN
16K
Jogo animado onde deverão ser apagados todos os pontinhos espalhados em um labirinto (o programa contém 15 tipos de labirinto). Você sera impedido a qualquer custo, por 4 extraterrenos, guardiães do labirinto, que poderão ser combatidos com cargas de raios-laser.
T-KALC
16K/64K
Programa desenvolvido para cálculos numericos em planilha. O usuario define as colunas, as linhas e as formulas aplicadas. Similar ao famoso Visicalc. De grande versatilidade, este programa permite a formulação de calculos cientificos e comerciais, analise de tabelas numericas e outras aplicações.
TKADREZ II
16K
Este jogo apresenta o tabuleiro e as peças no video. Permite a escolha de ate 7 niveis de dificuldade. O programa fornece a qualquer momento, a listagem dos lances efetuados, e armazena em fita a posição das peças. Ele podera recomendar a sua jogada.
NOS REVENDEDORES AUTORIZADOS EM TODO PAÍS
MICROSOFT®
80 programas à sua disposição. Solicite folheto.
Av. Angélica, 2.318 - 13º - Cj. 132
Tel.: (011) 256-3858 - CEP 01228 - São Paulo - SP

Expediente

PUBLICAÇÃO:
Micromega Publicações e Material Didático — Cx. Postal 60081
CEP 05096 — São Paulo — SP
DIRETOR:
Pierluigi Piazzi
JORNALISTA RESPONSÁVEL:
Aristides Ribas de Andrade Fº
ASSESSOR TÉCNICO:
Flavio Rossini
DIRETOR DE ARTE:
Fernando Leoni
ILUSTRAÇÕES:
Fatima M. Rossini Gouveia
ARTE E DIAGRAMAÇÃO:
Eliana Santos Queiroz,
Osmére Sarkis
COLABORARAM NESTE NÚMERO:
Igor Sartoni, Samuel Ejchel,
Dr. João Carlos Normanha Sales,
Jaime Maia Neto, Luis G.C. Antunes
CORRESPONDENTES:
Londres — Robert L. Lloyd
Paris — Alain Richard
N. York — Natan Portnoy
Milão — Licia Zappa
DEPARTAMENTO COMERCIAL:
Tel.: (011) 257-5767
PUBLICIDADE:
Lídia Pauluk, Mário Alves Vianna,
Aurio I. Mosolino
VENDA DE ASSINATURAS:
Carlos Henrique Oliveira
GERENTE COMERCIAL:
Gina Elimelek
ASSISTENTE COMERCIAL:
Atilio Debatin
PERIODICIDADE:
Mensal
TIRAGEM:
30.000 exemplares
NÚMERO 3:
Distribuição só para assinantes.

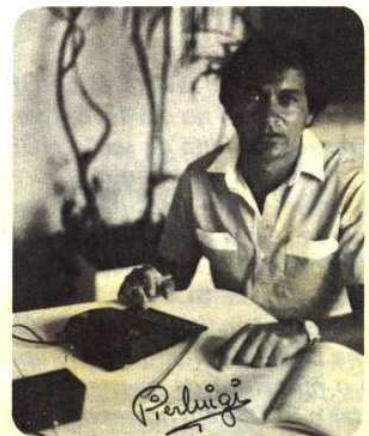
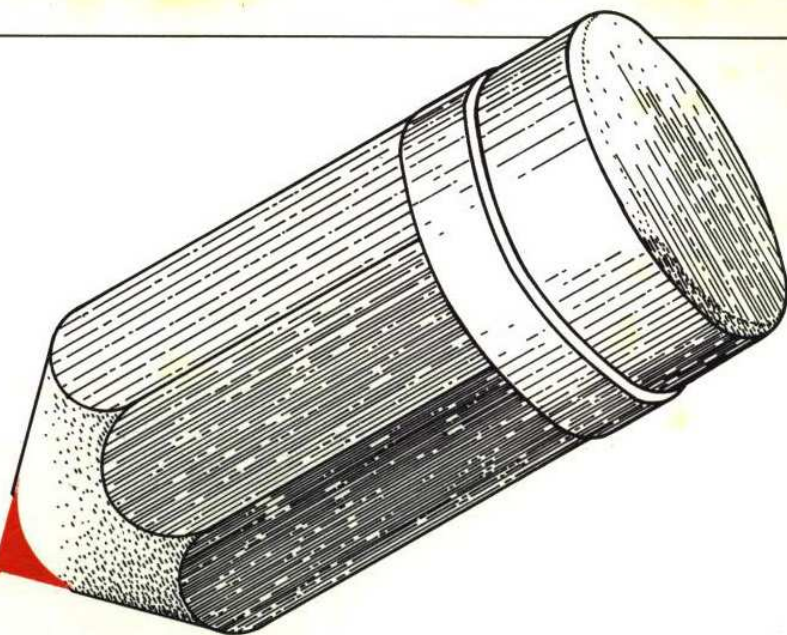
ASSINATURAS:
12 meses: Cr$ 11.800,00
Promoção especial até 30 de setembro:
Cr$ 9.900,00 com uma fita de brinde.

Índice

Matéria	pág.
Carta dos leitores	2
Mais traduções para o Basic TK	6
Matemática financeira em Basic	10
Como colaborar com Microhobby	12
Desgrilando	12
Programa do mês	14
Programa do mês	18
A fita do mês	25
Dicas	27
Dicas	29
Programa do Leitor	30
Programa do Leitor	31
Programa do Leitor	32
Novidades	35
Curso de Basic — Aula 3	38
Curso de Assembly — Aula 2	42
Quebra-Cabeça	44
Resposta do Quebra-Cabeça	46
Pequenos Anúncios	47
Como fazer sua assinatura	48
O que está reservado para o nº 4	48

Editorial

Em várias ocasiões já deixamos bem claro que o enfoque de nossa publicação é muito mais didático que jornalístico. Queremos, é lógico, que o usuário do TK seja alguém bem informado, mas também, e principalmente, queremos que ele **aprenda** *a tirar o máximo proveito de seu equipamento. O micro processador Z80, que é o coração do TK, tem funções apenas parcialmente aproveitadas por quem programa em BASIC. Por isso entramos com esse número, no universo do ASSEMBLY (incidentalmente, "ASSEMBLER" não é uma linguagem: é um programa que permite elaborar programas em ASSEMBLY).*

Como a linguagem de máquina é de árduo aprendizado, a nossa preocupação didática nos diz para não correr. Por isso, enquanto o curso de ASSEMBLY do Flavio Rossini avança, os artigos da revista neste tema se limitarão, por enquanto, a ensinar o leitor a digitá-los e utilizá-los sem entrar em detalhes sobre a linguagem em si.

Quanto ao crescimento (e não apenas físico) de nossa revista, este número o já fala por si. Só nos resta agradecer a avalanche de novos assinantes que apesar de alguns tropeços e atrasos iniciais (que vamos levar algum tempo para consertar) confiaram em nós e permitiram a consolidação deste empreendimento que abre um canal de diálogo, informação e **aprendizado** *para a comunidade cada vez mais vasta dos usuários do TK.*

CARTA DOS LEITORES

Caros amigos da Micromega,

Fiquei muito contente com o lançamento da revista "Usuários do TK". Era isso que faltava no mercado da informática, uma revista dedicada somente para nós, que possuímos um TK. Gostei muito dessa promoção de assinatura, no qual, ganhamos uma fita de programas "grátis". Desejo-lhes todo o sucesso.

Gostaria que me enviassem maiores informações sobre o programa "Barão Vermelho", o qual não rodou direito em meu TK.

Luciano Chaves (Florianópolis — SC)

Caro Luciano,

Obrigado pelo incentivo. Com relação ao programa "Barão Vermelho", que fez tanto sucesso que foi até copiado por outra revista do ramo, ele provavelmente não rodou direito no seu TK pois ele não possui **SLOW**. Para eliminar este inconveniente você deve inserir após cada instrução **PRINT** uma instrução **PAUSE** 3Ø. Desta forma a tela ficará piscando mas dará para jogar. Se você enviar o seu computador à fábrica da MICRODIGITAL, eles podem instalar o circuito **SLOW** para você. Antes escreva uma carta para saber o custo.

Prezados Senhores,

Ao folhear a revista MICROHOBBY, cujo primeiro exemplar recebi no dia 15/6, tive uma decepção.

Trata-se da falta da seção HOBBY indicada no exemplar nº Ø como a seção que apresentará projetos eletrônicos para serem executados por todos os leitores, que como eu têm a eletrônica como hobby. Nesta seção, deveria constar as informações necessárias para a adaptação da função ***SLOW*** *em nossos computadores. Cheguei a esta conclusão lendo o último parágrafo do programa Boliche editado no exemplar nº Ø, que informava que o número 1 da revista trazia os detalhes para acrescentar o* ***SLOW*** *em nossas máquinas.*

Gostaria também que me fosse sanada uma dúvida quanto ao recebimento da fita brinde, pois não a recebi juntamente com o exemplar nº 1. Pediria então que me informassem quando será feito o envio da referida fita.

Quero também parabenizá-los pelo conteúdo do nº 1 que pecou somente pelo reduzido número de páginas. Acho que seria de interesse de todos os leitores que a revista tivesse o número de páginas ampliado, abrindo novas seções, não esquecendo, portanto da seção HOBBY prometida no exemplar nº Ø, que não apareceu no nº 1 e que me parece não aparecerá no nº 2.

José Daniel Oliveira Vigário

Prezado José Daniel ,

Com relação à fita brinde houve realmente um atraso, motivado por várias coisas. Uma delas foi um "grilo" relativo ao TK85 que nos fez interromper a produção. Posteriormente, ela foi retomada a todo vapor e você ao ler estas linhas já deve tê-la recebido.

Com relação ao número reduzido de páginas, a revista que você têm em mãos já é uma resposta.

A seção HOBBY logo se desenvolverá a todo vapor. Recebemos de nossos correspondentes muitos projetos para "hobbystas" mas, infelizmente, nem sempre existem componentes nacionais disponíveis. Entretanto estamos fazendo as devidas adaptações para futura publicação.

Com relação ao **SLOW**, quando escrevemos ". . . como fazer para acrescentar **SLOW** à sua máquina." estávamos pensando em dar uma relação de oficinas aptas a fazer tal adaptação.

Tal relação ainda não foi publicada pois aguardamos respostas de uma série de cartas que enviamos para levantar tal lista.

Ficamos muito surpresos quando percebemos que alguns leitores estavam esperando o circuito.

Muito bem, agora vamos ter que montar uma matéria a esse respeito. Ela, porém, só poderá ser publicada se não violar segredos industriais das duas fábricas brasileiras envolvidas.

Prezados Senhores,

Antes de mais nada, os meus cumprimentos pelo lançamento de MICROHOBBY. Sem dúvida nenhuma, o TK é o computador mais vendido devido ao baixo custo (em relação aos outros micros existentes no mercado) e a sua grande capacidade de operação. Desejo muito sucesso a vocês, uma vez que nós, usuários do TK só temos a ganhar com isso.

Agora vamos às dúvidas, para as quais peço a atenção e elucidação de vocês:

1) Por que os "micros" nacionais são muito mais caros do que os originais estrangeiros? Seria talvez abuso da reserva de mercado?

*2) No nº Ø da MICROHOBBY, seção "Programa do Mês", no programa BOLICHE, vocês escreveram que no nº 1 seria mostrado como acrescentar "***SLOW***" aos TK's que ainda não o tivessem. Não mostraram e não comentaram nada a respeito. Então pergunto: como fazer para acrescentar "***SLOW***" ao meu TK?*

3) Como explicar o desempenho do programa "DEMOLIDOR", da MICROSOFT, num TK sem SLOW? Isto sempre me deixou bastante curioso.

4) Cadê a fita brinde? Nenhuma notícia sobre ela no nº 1.

5) É verdade que existem problemas com o joystick e a printer do TK?

Luis Henrique B. de Miranda – B. Horizonte – MG

Prezado Luis Henrique,

Obrigado pelos cumprimentos. Elucidando suas dúvidas pela ordem:

1) Se você fizer as contas pelo câmbio **real** e não oficial, verá que os micros nacionais não são tão mais caros. além disso, você deve saber que a "escala" de produção influi significativamente no custo. O Brasil é um país pobre, com um povo pobre e poucos privilegiados podem se dar ao luxo de comprar um micro: o TK que é o "proletário" dos micros custa 3 salários mínimos. Enquanto que no Brasil o número de micros que usam a lógica SINCLAIR ainda está longe de 50000, no mundo (Inglaterra, Itália, França, Estados Unidos e Canadá) temos 2 milhões.

2) Veja a resposta ao José Daniel

3) Passamos a pergunta para a seção DESGRILANDO (pág. 12).

4) Veja José Daniel

5) Os recentes "pacotes" fecharam a importação de uma peça vital para a impressora do TK. Enquanto perdurar esta proibição vamos ter que aguardar mais um pouco.

O Joystick não tem problemas. Quem pode apresentar problemas é um lote de máquinas que só funciona com joystick mediante a troca de alguns resistores. Se sua máquina tiver este problema fomos informados pela MICRODIGITAL que ela faz o ajuste citado gratuitamente, correndo por sua conta apenas as despesas de frete. Basta enviar sua máquina para a MICRODIGITAL, mesmo que ela já esteja fora de garantia.

Caro Senhor Diretor,

Tendo recebido o nº 1 da revista MICROHOBBY, tomo a liberdade de escrever-lhe esta curta missiva, pelos fatos que passo a expor:

*1) Sou possuidor de micro NEZ-8000, que, tal como os primeiros TK82 fabricados, não possui a modalidade "***SLOW***".*

2) Assim, examinando o nº Ø da revista, resolvi assiná-la, pois à página 6 (Programa BOLICHE) estava expressamente dito que no nº 1 constariam as informações de "como fazer para acrescentar o **SLOW** *à máquina."*

3) Qual não foi minha surpresa, ao verificar que no nº 1 não se fazia qualquer menção a tal assunto. Ora, entendendo que vários dos assinantes dessa revista, que como eu depositaram irrestrita confiança em sua publicação, (o que, cá entre nós, trata-se de risco incalculável, nos dias em que vivemos) são possuidores do NEZ-8000, acredito que a pretenção de muitas respostas não foi atendida.

4) É claro que se eu fosse adquirir um micro hoje, optaria pelos fabricados pela MICRODIGITAL, recusandome a adquirir qualquer produto da PROLÓGICA.

5) Assim, como tal não ocorreu, solicito de V.Sa. que corrija esta falha, publicando com urgência o esquema do **SLOW**.

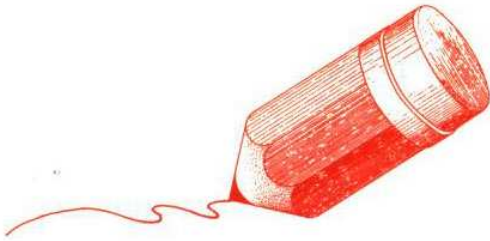
Como sugestão proporia que publicasse, também, esquema para se adaptar um ou dois joysticks (afinal a MICRODIGITAL tem que vendê-los, não?) no NEZ-8000 ou TK82C e modificação do fundo e caracteres no vídeo (de preto e branco, respectivamente no NEZ-8000 para branco e preto e vice-versa no TK82C).

Sérgio R.M. Ferreira – Santos/SP

Caro Sérgio

Calma! A maioria de suas dúvidas já foi esclarecida na resposta à carta do José Daniel. Apenas queremos acrescentar duas coisas:

1) Este tipo de cobrança é muito bom (mesmo que nós sejamos o alvo), pois o consumidor brasileiro não está acostumado a estrilar para fazer valer seus direitos.

2) A MICROMEGA, que edita a revista MICROHOBBY não é da MICRODIGITAL, apesar de manter um estreito relacionamento com esta firma, por estar publicando uma revista do interesse de seus clientes. O teor de sua

carta sugere uma interdependência que na realidade não existe. De qualquer forma passamos uma cópia de sua carta para a MICRODIGITAL, pois sua sugestão de 2 joysticks é bem interessante.

O

Prezados Senhores,

Com prazer, acuso o recebimento e agradeço o envio do nº 1 de MICROHOBBY, a revista dos Usuários do TK.

Valeu a pena esperar. A revista está sensacional em todos os aspectos. Uma curiosidade: alguém da Equipe é professor? Por favor, não estranhem a pergunta; é que eu sou, e a linguagem da Revista, tão didática, precisa e bem humorada, me soou "algo familiar"...

Assim sendo, somente posso aplaudir com entusiasmo o surgimento desta publicação, que tenho a certeza, muito virá acrescentar e a enriquecer tudo o que já existe entre nós sobre Informática.

Gostaria de saber se já estão disponíveis os livros anunciados e, se já estiverem, seus preços e condições de compra.

Aproveitando a ocasião, desejo submeter à consideração de V. Sas. duas dúvidas relacionadas a programas:

1) Em uma fita da Kristian chamada SPACE RAID, notei a presença de uma instrução **REM** *enorme (toma quase a tela toda), cheia de caracteres (aparentemente) sem sentido. O que pode ser isto?*

2) Outra fita, da MICROSOFT, chamada "Demolidor", apresenta problemas ao se tentar carregá-la: quase sempre, ao final do carregamento, a tela fica cheia de listas irregulares ou se apaga, obrigando, em ambos os casos, a desligar momentaneamente o computador, para que as coisas voltem ao normal, isto é, retorne o K. Se possível, gostaria de uma palavrinha sobre isso.

No mais, é aguardar (ansiosamente) o nº 2 e reiterar meus mais entusiásticos votos de crescimento e prosperidade para a Revista.

Seja bem vinda, MICROHOBBY!

Theodomiro Aguiar (Petrópolis — RJ)

Prezado Prof. Theodomiro,

Realmente na equipe temos dois professores, o PIERLUIGI e o FLAVIO, ambos extremamente bem humorados (apesar das fitas-brinde) e estamos em vias de aumentar a equipe com mais um. Isto não é mera coincidência ou uma espécie de "máfia pedagógica", mas faz parte da própria filosofia desta publicação: nossa finalidade não é apenas "informar", mas principalmente ensinar. Daí o cunho mais didático que jornalístico. Os livros anunciados estão em fase final de processamento gráfico e brevemente serão lançados no mercado.

Com relação às dúvidas sobre programas:

1) A gigantesca instrução **REM** na fita da Kristian armazena uma sequência de bytes que constitui uma subrotina em linguagem de máquina. A leitura de "LIFE" e "APAGANDO A TELA" publicados neste número esclarecerão melhor.

2) A fita deve estar com defeito ou sofreu, após sua produção, algum acidente magnético. Escreva à MICROSOFT, informando-se sobre a possibilidade de troca.

Finalmente, obrigado pelas boas-vindas.

O

Prezados Senhores,

Não gostei muito do 1º número da revista MICROHOBBY. Achei-a muito pequena e incompleta, apenas 16 páginas. A seção de DICAS, a meu ver, estava fraca e não achei nada sobre a seção HOBBY. Envio um programa de minha autoria na esperança de que a revista melhore, principalmente que ela cresça no tamanho. Trata-se do conhecido "Jogo da Forca" e ocupa quase 3 K bytes. Aproveitando esta envio algumas perguntas:
A impressora usada pelo TK82C é de boa qualidade?
Ela imprime todo tipo de símbolos gráficos?
Ela usa papel especial?
Como é feita a impressão?
O que significa eletro-sensível?
A impressora necessita de interface?
Há necessidade de alguma modificação no computador?
Se houver quanto custa e quanto tempo leva?

Fernando Araujo Lopes (São Paulo — SP)

Caro Fernando,

Como você pode ver a revista já cresceu bastante, graças a pessoas como você ou o Sérgio que apesar de não gostarem do começo e não confiarem muito, nos deram uma força e permitiram que a revista melhorasse cada vez mais.

Infelizmente não podemos aproveitar o seu "Jogo da Forca", pois as gavetas da redação transbordam com os "Jogos da Forca" já enviados. Queremos aproveitar a oportunidade para avisar o pessoal:

Forca
Senha
Velha
Invasores

não serão mais publicados, tá?!

Respondendo a suas perguntas:

A impressora usada pelo TK (por enquanto a versão ▷

inglesa, enquanto a brasileira tiver dificuldades de importação de peças) é de boa qualidade, imprime todos os símbolos que aparecem na tela e algo mais (veja "GRÁFICOS EM ALTA RESOLUÇÃO no nº 2). O papel é especial: ele é revestido por uma finíssima camada metálica sobre um fundo escuro.

Um estilete o percorre em alta velocidade. Como o papel é condutor de eletricidade, quando no estilete é aplicado uma tensão elétrica, dele salta uma faixa que faz o revestimento metálico se evaporar num ponto. Sob este ponto aparece o fundo escuro. Todas as cópias dos programas editados nesta revista são tiradas numa impressora deste tipo. O TK aceita esta impressora diretamente na conexão da expansão de memória. Como a conexão da impressora é do tipo macho de um lado e fêmea do outro, a expansão de memória pode ser ligada ao computador simultaneamente com a impressora. Já o CP 200 e o NEZ8000 da PROLÓGICA precisam de uma pequena adaptação, pois em sua fabricação foram utilizados conectores traseiros incompatíveis com os equipamentos SINCLAIR.

Temos notícias, inclusive, que dão conta de uma próxima fabricação de interfaces que permitirão conectar os TKs com as impressoras nacionais usando papel comum. Obviamente estas impressoras são muito mais caras.

Prezados Senhores,

Meu nome é Marcelo Arouca Araujo e tenho 14 anos. Eu sou fã de eletrônica e de computação, sei um pouco das duas. Quando comprei o meu computador, pensei em fazer um curso de BASIC, mas depois resolvi aprender sozinho, apenas olhando o manual que vem junto. Eu sei alguma coisa, mas isso não é suficiente ainda para poder fazer programas complicados, por isso eu mando para vocês este programa simples. Com o tempo eu espero melhorar e poder mandar programas à altura desta revista.

Eu gostei muito da primeira aula e das DICAS. Com a primeira aula da revista aprendi com mais detalhes o que eu já sabia, mais ou menos.

O programa que mando para vocês é muito famoso, chama-se Forca.

Caro Marcelo,

Se você der uma lida nas outras cartas verá que não publicamos Jogo da Forca. Apesar disto ficaremos muito felizes em receber outras colaborações suas. Mesmo programas "simples" podem ser muito úteis e neste caso serão impressos. Temos certeza, também, que logo logo você estará desenvolvendo coisas em nível "profissional".

Prezados Senhores,

Tudo muito bem, tudo muito ótimo. Mas será que não esqueceram algo? Se não me engano, eu deveria receber uma ou duas fitas justamente pelo fato de ser assinante desta (brilhante) revista.

A propósito: Será que os senhores poderiam enviar maiores detalhes sobre "TIG-LOADER"? (pág. 11 — nº 1), ou deveria escrever para tal firma?

Roberto Araujo (Rio de Janeiro — RJ)

Caro Dr. Roberto,

A fita é só **uma** (com **dois** jogos) e já foi enviada. Desculpe pelo atraso. Com relação ao TIG-LOADER, aqui na redação testamos um protótipo, mas não temos maiores detalhes técnicos ou comerciais. Acho que a solução é escrever para a firma.

Um abraço.

Caro Editor,

Recebi meu 1º número da revista Microhobby e gostei muito.

Aproveitei todos os programas, principalmente o utilitário. Gostaria de sugerir que fosse este transformado em linguagem de máquina.

Gostaria de saber em que livraria ou representante do TK, na região de Campinas, poderia encontrar as novidades anunciadas na página 12.

A única coisa que me entristeceu ao receber este número foi a falta de tão decantada fita com os jogos, por isso, peço ao senhor a gentileza de enviar-me uma fita, pois não gostaria de começar mal a minha coleção.

Cláudio S. Tonhetta (Sumaré — SP)

Prezado Cláudio,

Os livros anunciados serão lançados brevemente, pois estão na fase final de processamento. A fita já foi enviada e com relação à transformação do programa do BASIC para ASSEMBLY temos uma sugestão. Estude com cuidado as lições que estamos publicando e logo você estará em condições de fazer isto sozinho. Quando isto acontecer publicaremos seu programa com muito prazer (e muito orgulho).

MAIS TRADUÇÕES PARA O BASIC TK

Igor Sartoni

49 Kisomo 30 cts
Gĩthomo Mũrang'a kĩa andũ
kibaru
Game su
Dunyagbləha RPT xə fè ewo

No número anterior de nossa revista falamos sobre a instrução **IF** . . . **THEN** . . . **ELSE** . . . , comum a vários computadores, mas que, por razões construtivas, não foi incluída no conjunto de instruções do TK. Todavia, com um pouco de imaginação, conseguimos contornar o problema. Sabemos que esta não é a única instrução que não faz parte dos recursos do TK e de computadores similares. Além disso, é comum em alguns computadores, como, por exemplo, no TRS-80, o uso de instruções **IF** . . . **THEN** . . . **ELSE** encadeadas, o que pode dificultar a **tradução** de um programa para outro.

Condicionais Encadeadas

Chamamos de condicionais encadeadas duas ou mais instruções **IF** . . . **THEN** . . . **ELSE** . . . dispostas de tal forma que uma condicional permite ou não o acesso a uma outra condicional, com a seguinte estrutura:

IF$_1$ (condição 1) **THEN IF**$_2$ (condição 2) **THEN IF**$_n$ (condição n) **THEN** (função ou operação n + 1) **ELSE**$_n$ (função ou operação n) . . . **ELSE**$_2$ (função ou operação 2) **ELSE**$_1$ (função ou operação 1).

Por exemplo:

1Ø **IF** X = Ø **THEN IF** Y = Ø **THEN IF** Z = Ø **THEN LET** A = Ø **ELSE LET** A = 1 **ELSE LET** B = 1 **ELSE LET** C = 1

Este tipo de estrutura pode ser compreendido como uma operação matemática com diversos níveis de parenteses. Assim, a primeira condicional **IF** "fecha" com a última condicional **ELSE**, a segunda com o penúltimo, e assim por diante.

Levando isso em consideração, podemos "quebrar" esta linha de programa em três partes:

1) **IF** X = Ø **THEN** **ELSE LET** C = 1

2) **IF** Y = Ø **THEN** **ELSE LET** B = 1

IF Z = Ø **THEN LET** A = Ø **ELSE LET** A = 1

▷

Se você observar bem, o que isto pode significar é que podemos transformar esta linha de programa em outras três, desde que consigamos preencher o espaço em branco entre as palavras **THEN** e **ELSE**.

Para preencher este espaço, devemos nos lembrar o que uma instrução desta natureza faz. A primeira coisa que devemos levar em conta é que, nesta linha de programa, se a condição X = Ø for satisfeita, a condição Y = Ø deve ser verificada. Do mesmo modo, se a condição Y = Ø for satisfeita, deve-se verificar a condição Z = Ø.

Assim, o que devemos fazer após verificar uma dada condição é verificar a condição seguinte. Levando isso em conta, a linha de programa em questão pode ser reescrita da seguinte forma:

```
1Ø IF X = Ø THEN GOTO 2Ø ELSE LET C = 1

2Ø IF Y = Ø THEN GOTO 3Ø ELSE LET B = 1

3Ø IF Z = Ø THEN LET A = Ø ELSE LET A = 1

4Ø . . . (sequência do programa).
```

Existe um pequeno problema, que deve ser contornado: ao completarmos uma linha em um programa em BASIC, o computador executará imediatamente a linha seguinte. Entretanto, a linha seguinte contém uma comparação que só deve ser feita se e somente se a comparação anterior for verdadeira. Da forma em que se encontra a nossa solução, se a primeira condição for verdadeira, a segunda condição é verificada. Se for falsa, C é igualado a 1 e a segunda condição é verificada, coisa que não queremos. Para que isso não ocorra, devemos desviar da linha que contém a comparação seguinte por meio de um desvio incondicional. Assim:

```
1Ø  IF X = Ø THEN GOTO 2Ø ELSE LET C = 1

15 GOTO 4Ø
2Ø  IF Y = Ø THEN GOTO 3Ø ELSE LET B = 1

25 GOTO 4Ø
3Ø IF Z = Ø THEN LET A = Ø ELSE LET A = 1

4Ø . . . (sequência do programa)
```

Resta mais um problema: a condicional **ELSE**. Para resolver este problema, recorreremos a uma das soluções que indicamos no artigo que publicamos no número anterior desta revista. Assim, por exemplo, o trecho onde aparecem as linhas 1Ø e 15 ficará:

```
1Ø IF X = Ø THEN GOTO 2Ø
12 IF X <> Ø THEN LET C = 1
15 GOTO 4Ø
```

Da mesma forma, poderemos aplicar esta solução para as outras linhas do programa, onde aparecem condicionais **ELSE.**

INPUT múltiplo e com mensagens

Algumas vezes encontramos, em alguns programas desenvolvidos para outros computadores, **INPUT** em que são pedidas mais de uma variável e, além de tudo, aparece uma mensagem qualquer, por exemplo:

```
1Ø INPUT "ENTRE COM OS VALORES DE A E B", A,B
```

Quando aparecer um **INPUT** com mensagens, o computador primeiro imprimirá a mensagem e ficará aguardando a introdução dos valores. No nosso exemplo, são apenas dois valores. Devemos digitar o primeiro valor pedido, uma vírgula e o segundo valor pedido, e, então, digitar **ENTER**.

No TK e em computadores semelhantes, tanto há mensagem como várias variáveis de entrada não podem ser digitadas na mesma linha. A solução é decompor esta instrução em várias outras, levando em consideração também que, numa instrução **INPUT**, em alguns computadores diferentes do TK, uma vez digitada a tecla **ENTER**, o valor digitado é impresso imediatamente.

Tendo isto em mente, podemos reescrever a linha em questão da seguinte maneira:

```
2 PRINT "ENTRE COM OS VALORES DE A E B"
3 INPUT A
4 PRINT A
5 INPUT B
1Ø PRINT B
```

Nem a disposição das variáveis nem a mensagem precisam ser rejeitadas. O melhor que devemos fazer é adaptar o trecho do programa às nossas necessidades e com o nosso gosto. Assim, poderemos, por exemplo, dar a seguinte forma ao trecho de programa em questão:

```
1Ø PRINT "QUAL O VALOR DE A?"
2Ø INPUT A
3Ø PRINT A
4Ø PRINT "QUAL O VALOR DE B?";
5Ø INPUT B
6Ø PRINT B
```

▷

ou ainda:

```
10 PRINT "A = ",
20 INPUT A
30 PRINT A
.
.
.
```

ON A GOTO nº de linha 1, nº de linha 2, nº de linha 3 . . . nº de linha n.

A instrução **ON A GOTO** tem como função enviar o programa para uma entre várias alternativas dadas por uma condição, representada por uma variável inteira, cujo nome é escrito por uma única letra (por exemplo, K). Por exemplo, na linha de programa:

```
10 ON C GOTO 100, 230, 350
```

Se C for 1, o computador vai executar a linha 100; se for 2, 230 e, se 3, 350. Se C for menor do que 1 ou maior do que n (no caso, 3), o programa vai para a linha seguinte. Os números de linha não precisam estar em sequência. Assim:

```
100 ON G GOTO 400, 10, 350
```

Se G for 1, o programa irá para a linha 400, se for 2, para linha 10 e se for 3, para linha 350.

Se você quiser usar este recurso em seu TK você poderá usar um dos "macetes que mostraremos a seguir".

1º caso: o programa original apresenta os números de linha após o GOTO numa sequência crescente de múltiplos de um mesmo número inteiro.

Suponha que o programa que você quer "traduzir" para seu computador tenha a seguinte linha:

```
50 ON A GOTO 100, 200, 300
```

Note que os números de linha estão numa sequência crescente de números inteiros, todos múltiplos de 100. Para este caso, podemos usar:

```
50 GOTO 100 * A
60 . . . (sequência do programa)
```

Todavia, a instrução **ON A GOTO** . . . faz ainda outra coisa: se **A** for menor que 1 ou maior que 3, o computador executará a linha seguinte. Para completar o trabalho, devemos acrescentar a linha:

```
45 IF A < 1 OR A > 3 GOTO 60
```

2º caso: o programa original apresenta quaisquer números de linha após o GOTO.

Quando o programa original apresenta quaisquer números de linha após o **GOTO** só podemos lançar mão de condicionais do tipo **IF . . . THEN**. Isto é possível porque o A ou a variável que indica o desvio pode ser encarada como uma variável que deve atender a uma dentre n condições. Por exemplo, a linha:

```
50 ON J GOTO 80, 10, 70
```

Poderá ser escrita como:

```
50 IF J = 1 THEN GOTO 80
55 IF J = 2 THEN GOTO 10
57 IF J = 3 THEN GOTO 70
60 . . . (sequência do programa)
```

Neste caso, não é necessário acrescentar nenhuma linha referente à comparação com valores menores que 1 e maiores que n, pois o programa segue automaticamente para a linha seguinte, após passar por todas as comparações.

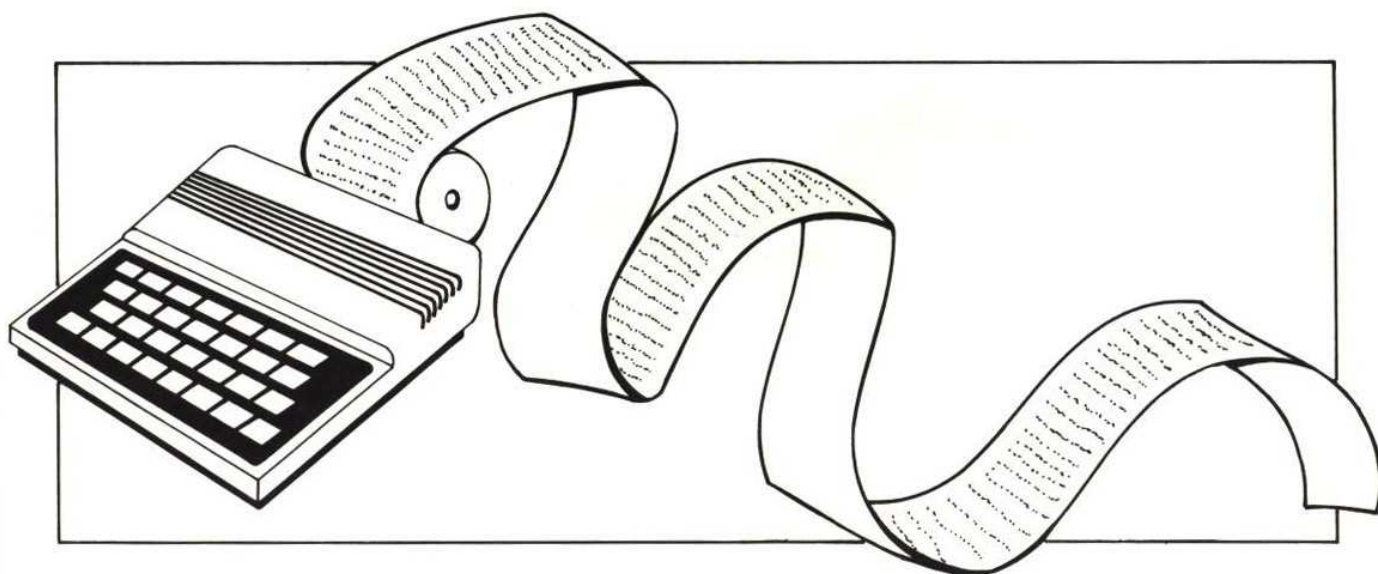
Conclusão

Existem muitos outros problemas encontrados por quem deseja rodar um programa de outro computador no TK ou em computadores com a mesma lógica. Ainda existem outras instruções que podem ser encontradas em outros computadores e não no TK (ou encontradas sob uma sintaxe diferente) e muitas delas poderão ser facilmente implementadas no seu computador. A recíproca também é verdadeira. Existem funções que aparecem no TK e não aparece em outros computadores, que obrigam seus usuários a lançar mão de outros recursos para contornar o problema. Um exemplo típico é a função **PAUSE** que normalmente não é encontrada em outros computadores.

Isso não torna o BASIC do TK inferior ao dos demais computadores, pois cada fabricante deu uma prioridade diferente ao seu projeto. No caso do TK, a orientação foi torná-lo compacto e a um preço acessível, com um hardware mínimo. O resultado foi uma linguagem BASIC um pouco diferente das demais, contendo instruções completas em cada tecla, ocupando um pequeno espaço na memória de programa.

Por outro lado, esta série de artigos será aberta à sugestões. Se você deparar com um programa de um outro computador que não conseguir traduzir, envie-nos uma cópia, citando a marca e o modelo do computador e a instrução que lhe causou problemas. Se for possível uma tradução, a instrução em questão poderá ser objeto de um futuro artigo. ○

MATEMÁTICA FINANCEIRA EM BASIC TK

Flavio Rossini

Muitas pessoas se lamentam da falta de comandos específicos em BASIC que tornem as "entradas" e "saídas" de dados numéricos compatíveis com o padrão utilizado em aritmética financeira, ou seja, números encostados à direita (e não à esquerda), com duas casas depois da vírgula (e não ponto) e com um ponto entre cada 3 algarismos. De fato, é desagradável ver os "milhões" na mesma coluna dos "milhares" ou então ver dados com milésimos de cruzeiros.

Cr$ 1934.8769
Cr$ 7.9815
Cr$ 299873.4
Cr$ 9
Cr$ 10.056

Apesar de outras linguagens como o FORTRAN, e principalmente o COBOL, fornecem facilidades para o manuseio de números nestes formatos, em BASIC isto também é possível. . .

.Para tanto, desenvolvi duas subrotinas em BASIC para o TK que executam, respectivamente, a "entrada" e "saída" de dados no padrão financeiro. Analizemos, primeiramente a entrada de dados; ela deve aceitar números com "." e ",", por exemplo:

1.345.003,37

e transformá-lo para a maneira que o computador está "acostumado" para a execução de cálculos, ou seja:

1345003.37

Assim temos:

```
 990 REM **FLAVIO ROSSINI 1983**
 995 REM SR P/ENTRADA DE DADOS
    C/ FORMATO FINANCEIRO
1000 FAST
1005 INPUT A$
1010 LET L=LEN A$
1015 DIM B$(L)
1020 FOR U=1 TO L
1025 IF A$(U)<>"." THEN LET B$(U
)=A$(U)
1030 IF A$(U)="," THEN LET B$(U)
="."
1035 NEXT U
1040 LET N=VAL B$
1050 RETURN
```

fig. 1

Esta subrotina simplesmente retira todos os "." do dado entrado e substitui a ",", por um ".", deixando o valor numérico para cálculos na variável N.

A subrotina para executar a "saída" de dados, ou seja, transformar um número no formato do computador para o formato financeiro, é bem mais complexo. Ela consta de três módulos básicos:

a) ajustar o número para duas "casas" após o " " (que ▷

será substituído por uma vírgula) para os três casos possíveis: nenhuma casa após o "." (variável inteira), uma casa após o "." e duas ou mais casas após o ".".
b) colocar um "." entre cada 3 algarismos antes da vírgula (",").
c) imprimir o número encostando-o à direita na vigésima coluna.

```
1090 REM **FLAVIO ROSSINI 1983**
1095 REM SR P/SAIDA DE DADOS
      C/FORMATO FINANCEIRO
1100 FAST

1105 REM MODULO A
1110 LET N$=STR$ N
1115 LET L=LEN N$
1120 FOR U=1 TO L
1125 IF N$(U)="." THEN GOTO 1140
1130 NEXT U
1135 LET N$=N$+",00"
1140 LET N$(U)=","
1145 LET N$=N$+"0"
1150 LET L=U+2
1155 LET N$=N$( TO L)

1160 REM MODULO B
1165 LET L=L-3
1170 LET L=L-3
1175 IF L<=0 THEN GOTO 1210
1180 LET C$=N$( TO L)
1185 LET N$(L)="."
1190 LET N$=N$(L TO )
1195 LET N$=C$+N$
1200 GOTO 1170

1205 REM MODULO C
1210 PRINT "CR$ ";TAB (21-L);N$
1220 SLOW
1225 RETURN
```

fig. 2

O número a ser colocado deve ser colocado na variável N. A título de exemplo, vamos apresentar um pequeno programa que utiliza estas subrotinas; você fornece o preço de custo de 3 produtos e a porcentagem de lucro desejada para cada um e ele fornece o preço de venda e o lucro . . .

```
   5 REM **FLAVIO ROSSINI 1983**
  10 REM EXEMPLO DE APLICACAO PA
RA  SUBROTINAS FINANCEIRAS
  20 SLOW
  25 LET ENTFIN=1000
  30 LET SAIFIN=1100
  35 DIM P$(3,14)
  40 DIM K(3,2)
  45 FOR I=1 TO 3
  50 PRINT "NOME DO PRODUTO ";I;
?"
  55 INPUT P$(I)
  60 PRINT P$(I)
  65 PRINT "PRECO DE CUSTO ?"
  70 GOSUB ENTFIN
  75 LET K(I,1)=N
  80 PRINT "CR$ ";A$
  85 PRINT "PORCENTAGEM DE LUCRO
?"
  90 GOSUB ENTFIN
  95 LET K(I,2)=N
 100 PRINT A$;" POR CENTO"
 105 PAUSE 120
 110 CLS
 115 NEXT I
 120 FOR I=1 TO 3
 125 LET LUCRO=(K(I,2)*0.01)*K(I
1)
 130 PRINT "PRODUTO ";P$(I)
 135 LET N=K(I,1)+LUCRO
 140 PRINT TAB 5;"PRECO DE VENDA

 145 GOSUB SAIFIN
 150 LET N=LUCRO
 155 PRINT TAB 5;"LUCRO"
 160 GOSUB SAIFIN
 165 PRINT
 170 NEXT I
 175 PRINT
 180 PRINT "BFE"
 190 STOP
```

```
EXEMPLO:

NOME DO PRODUTO 1 ?
ABACAXI
PRECO DE CUSTO ?
CR$ 560
PORCENTAGEM DE LUCRO ?
71,87 POR CENTO



PRODUTO ABACAXI
    PRECO DE VENDA
CR$                   962,47
    LUCRO
CR$                   402,47

PRODUTO BANANA
    PRECO DE VENDA
CR$                   619,90
    LUCRO
CR$                   262,40

PRODUTO CAVIAR
    PRECO DE VENDA
CR$                27.268,37
    LUCRO
CR$                18.280,77

BFE
```

fig. 3

Note que as subrotinas utilizadas apenas "truncam" as duas casas depois da vírgula; se for necessário um "arredondamento" do resultado, bastaria, por exemplo, antes de entrar no módulo B da subrotina de saída, restar o **valor** do caractere N$ (L+1); se **VAL** N$ (L+1) for maior do que 5, basta somar Ø,Ø1 ao resultado; caso contrário, nenhuma providência deverá ser tomada. Tente fazer isto. É um bom exercício e o ajudará a entender as subrotinas aqui apresentadas.

COMO COLABORAR COM MICROHOBBY

A revista MICROHOBBY foi criada para servir de intercâmbio entre os leitores que participam do mágico mundo da computação.

A característica realmente inovadora do computador pessoal, está em transformar cada consumidor num criador. Aproveite sua criatividade e envie suas colaborações, recebendo remuneração a título de DIREITO AUTORAL.

Se o material enviado for aprovado para publicação, a remuneração será de Cr$ 10.000,00 por programa. Se o programa for muito bom e muito extenso esta quantia será aumentada, a critério da redação, até Cr$ 30.000,00. Esta remuneração será oferecida também para artigos interessantes sobre o mundo do TK.

A maneira ideal de nos enviar o material a ser publicado obedece às seguintes normas:

1) **Nunca** esquecer de colocar nome completo, endereço e número de sua assinatura em **todo** material enviado (fitas, listagens de impressora, envelope, carta, etc).
2) Enviar a listagem de programa **datilografada** ou, melhor ainda, tirada na impressora do TK.
3) Colocar sempre uma linha **REM** com o nome do autor e o título do programa.
4) Enviar uma fita com o programa gravado **algumas vezes** (se possível em gravadores diferentes).
5) Na fita, gravar **com microfone** (em viva voz), algumas instruções úteis, **nome completo e endereço do remetente.**

O material não utilizado não será devolvido. Esta é uma norma corrente em quase todas as revistas do mundo, que gera porém, uma certa desconfiança: "e se apagarem meu nome do programa e o publicarem sem me remunerar?"

Neste ponto podemos sugerir o seguinte:

Ao enviar o material para **qualquer** revista, você, ou confia em sua seriedade, ou, é melhor não enviar o material, reservando-o para uma publicação dígna de seu crédito.

Uma outra saída é a de tirar uma cópia de todo material enviado e registrá-lo em cartório, para qualquer reclamação posterior. Obviamente tal procedimento só é aconselhável se o material enviado for valioso a ponto de justificar toda essa mão-de-obra.

6) O material deve ser enviado para:

MICROMEGA P.M.D. Ltda.
PROGRAMAS DO LEITOR
Cx. POSTAL: 60081
CEP: 05096 – S. PAULO – SP

7) Qualquer dúvida, poderá ser esclarecida pelo telefone:

(011) 257-5767 com Roberto.

Aguardamos sua colaboração.

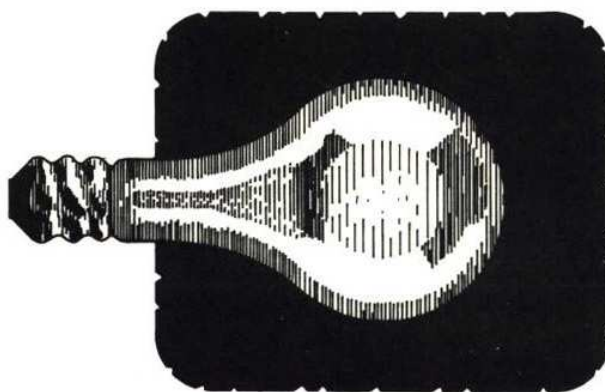
DESGRILANDO

Prezados Senhores,

Se há um programa na memória do computador e eu dou LOAD, para procurar um programa numa fita. O programa anterior se apagará? Parcial ou totalmente? O programa da fita se carregará automáticamente?

Jaime Jacobson

Caro Jaime,

Ao dar o **LOAD** o programa que está na **RAM** é apagado totalmente para dar lugar ao novo. Se o novo não estiver em boas condições você fica com o computador vazio.

Nisso é que reside a grande vantagem da função **VERIFY** do TK85: após gravar uma fita você entra com o programa, mas sem dar um **LOAD**. Se o programa gravado conferir com o original da **RAM** o computador acusará OK e você poderá desligá-lo tranquilamente.

O programa da fita carrega-se automaticamente se você digitar seu nome. Por exemplo, se você tiver vários programas gravados numa única fita e der o comando:

LOAD "FORCA"

Com o gravador na posição **PLAY**, a fita correrá sem que o gravador aceite os programas que passam. Ao receber, porém, uma sequência correspondente a um programa que foi gravado por um:

SAVE "FORCA"

o computador carregará sua **RAM** com ela.

Prezados Senhores,

No programa Barão Vermelho, da página nº 5, da Revista dos Usuários do TK nº Ø, há a linha 18Ø que usa as funções **PEEK**, *ou seja: 18Ø* **IF PEEK** *(***PEEK** *16398 + 256 X* **PEEK** *16399) = 128* **THEN GOTO** *31Ø. Qual é o significado destas instruções?*

Na linha 215 e 24Ø trabalham com **SGN** *que é outra dúvida.*

Eulálio Alves da Silva (Porto Velho – RO)

Caro Eulálio,

Os endereços 16398 e 16399 contém dois bytes que indicam uma variável do sistema denominada **POSPR**. Esta variável, por sua vez, indica a posição do **PRINT** no arquivo de imagem. Para uma explicação detalhada, porém, seria interessante que você acompanhasse uma matéria a ser publicada no número 4, justamente sobre a organização da memória do TK.

Por outro lado, a função **SGN** (tecla F) é fácil de ser explicada. Ela assume o valor +1 se seu argumento é um número positivo qualquer, −1 se for negativo e Ø se for nulo.

Experimente fazer no seu computador o seguinte programinha:

```
10 DIM A$(3,10)
20 LET A$(1)="SGN 89"
30 LET A$(2)="SGN 0"
40 LET A$(3)="SGN -55"
50 FOR I=1 TO 3
60 PRINT A$(I);"=";VAL A$(I)
70 NEXT I
```

Ele vai lhe explicar o **SGN**.

Prezados Senhores,

. . . Como explicar o desempenho do programa "DEMOLIDOR" da MICROSOFT num TK **sem SLOW?** *Isto sempre me deixou bastante curioso. . .*

Luis Henrique B. de Miranda

Caro Luis Henrique,

A explicação é simples: um computador que trabalha em **FAST** não envia dados para a TV enquanto está processando, por isso "perde o controle" de imagem. Um computador com circuito de **SLOW** usa este circuito como uma espécie de interruptor: enquanto a tela está sendo formada pela varredura, o computador é obrigado a não processar, mantendo a formação da imagem sob controle. No breve intervalo de tempo entre o fim de uma imagem e o começo da seguinte há um lapso de tempo durante o qual ele processa os dados pois neste intervalo não há problema em não se enviar dados para a TV. O programa DEMOLIDOR é uma obra prima da acrobacia de programação: como cada instrução em linguagem de máquina tem um tempo de execução muito bem determinado, a sequência de instruções foi calculada de sorte a durar exatamente tanto quanto o lapso citado. Ao terminar este lapso, o próprio programa faz o micro processador parar de processar durante a formação de imagem. Note que, como o programa tem ramificações, este cálculo foi feito para todas elas.

Em resumo, "DEMOLIDOR" é um programa que trabalha aos 'soluços' em perfeita sincronia com a varredura da TV, simulando um **SLOW** por "software".

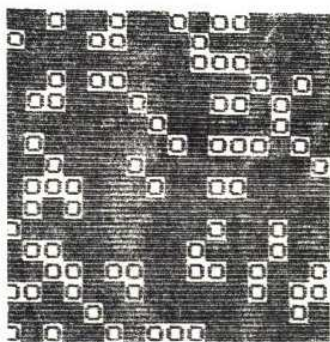
Programa do Mês
"LIFE"
Simulação da Evolução

Pierluigi Piazzi

O programa "**LIFE**" foi originalmente elaborado por John Conway da Universidade de Cambridge, Inglaterra em 1970.

A versão que apresentamos foi adaptada para o TK por Flavio Rossini e encontra-se detalhada em seu livro "Introdução à Linguagem de Máquina para o TK".

Vamos nos limitar a reproduzir aqui a listagem do programa sem entrar na discussão das instruções de seu Assembly. Se algum leitor estiver interessado nesta parte, aconselhamos a leitura do capítulo XI do livro citado.

Inicialmente vamos discutir o princípio do programa: num quadrado de 16x16 posições, nascem, se reproduzem e morrem células, representadas por um "O" (fig. 1).

EXEMPLO DE TELA

1053 BYTES

fig. 1

Ao começar o programa produz uma primeira geração de células, distribuidas ao acaso em seu "habitat" de 256 posições. A geração seguinte é então desenhada na tela obedecendo a certas regras "biológicas".

Cada célula tem 8 vizinhas possíveis (fig. 2):

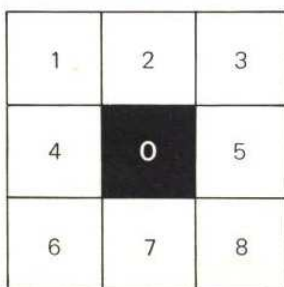



fig. 2

1) As células que tiverem apenas 2 ou 3 vizinhas sobrevivem e aparecem na geração seguinte. As demais morrem.

2) Todo espaço vazio que tiver exatamente 3 células vizinhas, gerará uma nova célula na geração seguinte.

É interessante observar que o "habitat" é um espaço fechado sem limites, ou seja, a última fila embaixo é computada como adjacente à primeira fila de cima e à última coluna da direita é considerada adjacente à primeira da esquerda (fig. 3).

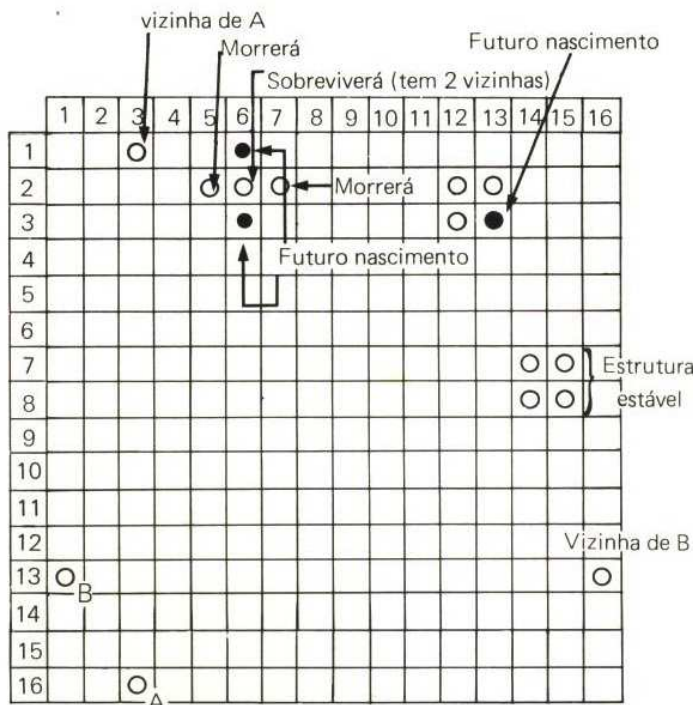



fig. 3

Em matemática, diríamos que o quadrado gerado na tela é topologicamente equivalente a uma esfera. Para entender melhor a analogia, você deve imaginar um mapa-mundi (fig. 4), onde o lado direito junta-se a seu lado esquerdo.

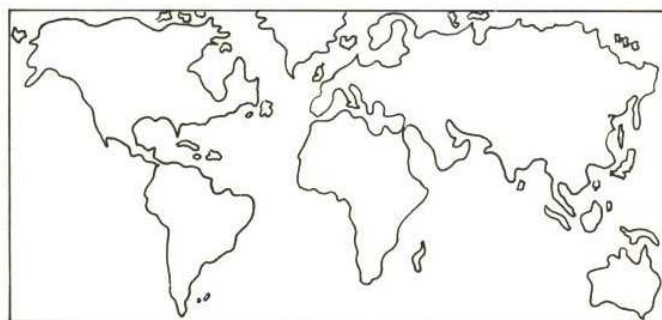



fig. 4

Qualquer outra observação a respeito da "biologia" do programa torna-se ineficaz enquanto você não tiver digitado o programa que, em sua forma final, ocupa 1Ø53 bytes.

1º Passo: digite o programa listado na figura 5, tomando o cuidado de entrar com exatamente 142 pontos após o **REM** da linha 1. Por via das dúvidas salve-o em fita para não ter que redigitar tudo se faltar energia elétrica. O programa em BASIC serve para inserir, no lugar de cada ponto, um valor correspondente ao código de máquina. O primeiro ponto após o **REM** está no endereço 16514 da memória, e o último no 16655. Esta inserção é feita através do comando **POKE**.

```
PROGRAMA 1

   1 REM ........................
..................................
..................................
..................................
......................
   5 SCROLL
  10 PRINT "ENDERECO INICIAL ? "
;
  20 INPUT M
  30 PRINT M
  40 FOR I=M TO 16655
  50 INPUT P
  60 POKE I,P
  70 SCROLL
  80 PRINT I,P
  90 NEXT I
```

fig. 5

2º passo: com o programa 1 já digitado, dê um **RUN**. O programa perguntará qual o endereço inicial, e você digitará 16514.

A seguir a sucessão de bytes, escritos em forma decimal, listados na figura 6, deverá ser digitada (apertando-se **NEW LINE** após cada byte). Esta é a parte mais chata da história e aconselhamos leitor a achar um colaborador disposto a ditar a sequência e a conferir na tela.

Lembra-se a Lei de Murphy?

"Se alguma coisa tem a mais remota chance de dar errado, fatalmente dará errado". Consequentemente, em algum ponto desta enfadonha sequência você errará. Não entre em pânico.

Procure o último endereço em que você digitou corretamente e anote-o. Digite **STOP** interrompendo o programa e dê um **RUN**.

Neste momento parece que estou vendo a cara desconfiada do leitor: "Tem certeza que é **RUN**? Não vou perder tudo que digitei até agora?"

Não, não vai perder: o **POKE** da linha 6Ø está reescrevendo a linha 1, substituindo os pontos por caracteres correspondentes aos bytes do programa em linguagem de máquina. O **RUN** apaga as variáveis, mas não as linhas do programa.

```
LISTAGEM DECIMAL DE "LIFE"
16514=>239
16515=>1
16516=>1
16517=>16
16518=>16
16519=>255
16520=>255
16521=>240
16522=>14
16523=>16
16524=>6
16525=>16
16526=>42
16527=>50
16528=>64
16529=>84
16530=>93
16531=>41
16532=>41
16533=>25
16534=>41
16535=>41
16536=>41
16537=>25
16538=>34
16539=>50
16540=>64
16541=>124
16542=>254
16543=>196
16544=>56
16545=>4
16546=>62
16547=>180
16548=>24
16549=>2
16550=>62
16551=>128
16552=>215
16553=>16
16554=>227
16555=>62
16556=>118
16557=>215
16558=>13
16559=>32
16560=>219
16561=>201
16562=>6
16563=>0
16564=>17
16565=>0
16566=>117
16567=>42
16568=>12
16569=>64
16570=>229
16571=>126
16572=>35
16573=>254
16574=>127
16575=>56
16576=>250
16577=>18
16578=>19
16579=>16
16580=>246
16581=>17
16582=>0
16583=>117
16584=>213
16585=>14
16586=>0
16587=>209
16588=>225
16589=>126
16590=>254
16591=>127
16592=>48
16593=>3
16594=>35
16595=>24
16596=>246
16597=>229
16598=>235
16599=>229
16600=>17
16601=>130
16602=>64
16603=>26
16604=>254
16605=>14
16606=>40
16607=>11
16608=>19
16609=>133
16610=>111
16611=>126
16612=>254
16613=>180
16614=>32
16615=>243
16616=>12
16617=>24
16618=>240
16619=>225
16620=>121
16621=>254
16622=>2
16623=>56
16624=>15
16625=>254
16626=>4
16627=>48
16628=>11
16629=>254
16630=>3
16631=>40
16632=>3
16633=>126
16634=>24
16635=>6
16636=>62
16637=>180
16638=>24
16639=>2
16640=>62
16641=>128
16642=>227
16643=>119
16644=>35
16645=>227
16646=>35
16647=>229
16648=>125
16649=>167
16650=>32
16651=>189
16652=>225
16653=>225
16654=>201
16655=>0
```

fig. 6

Ao ser perguntado qual o endereço inicial digite o sucessivo ao que você anotou e continue inserindo a lista.

Quando aparecer o código:

Ø/9Ø

Você pode ofertar **NEW LINE** e verá como a linha 1 mudou de aspecto: uma parte, inclusive, ficou invisível (fig. 7).

```
ASPECTO DA LISTAGEM APOS RODAR
O PROGRAMA 1

   1 REM LOAD ▀▀ (( COPY COPY LIS
T : (▞(EMRND??DD;DDD;6MRND? RETUR
N CODE S▄Y█/ ▀Y█NOT ( STOP Y




OBS: NUNCA COMANDE "EDIT" PARA
  A LINHA 1
```

fig. 7

Nunca, mas **nunca mesmo,** traga esta linha 1 para baixo com o comando **EDIT**. Se você acompanhar o nosso curso de Assembly, um dia verá o porquê.

3º passo: apague todas as linhas do programa 1 exceto a número 1, que contém a sequência em linguagem de máquina. Se você ficar aflito por apagar às cegas, digite **LIST** 2.

```
PROGRAMA 2

  2█SLOW
  3 RAND
  4 LET MICROHOBBY=USR 16522
  5 LET MICROMEGA=USR 16562
  6 GOTO 5
```

fig. 8

Agora digite o programa 2 (fig. 8). Se seu computador não tiver **SLOW**, digite o programa 2 na versão apresentada na figura 9.

```
PROGRAMA 2 P/QUEM TEM MAQUINA
SEM "SLOW"

  3 RAND
  4 LET MICROHOBBY=USR 16522
  5 LET MICROMEGA=USR 16562
  6 PAUSE 100
  7 GOTO 5
```

fig. 9

A linha 4 chama, através da função **USR** a subrotina localizada na linha **REM** a partir do endereço 16522. Esta subrotina gera aleatoriamente a primeira geração de células (fig. 1). A seguir a linha 5 chama a subrotina que se inicia no endereço 16562.

Esta subrotina impõe as "regras biológicas" para produzir a geração seguinte.

4º passo: digite **RUN** e fique observando o suceder-se de gerações. Quem não tem **SLOW** talvez tenha que ajustar um pouco o controle **VERTICAL** de sua TV: a tela ficará piscando mas não pulando.

Se você quiser acompanhar passo a passo a "evolução" de seus bichinhos, pode alterar o programa 2 como indicado na figura 1Ø. Desta forma, para ver a geração seguinte basta pressionar **NEW LINE**.

```
  2█SLOW
  3 RAND
  4 LET MICROHOBBY=USR 16522
  5 LET MICROMEGA=USR 16562
  6 INPUT A$
  7 GOTO 5

PROGRAMA PASSO-A-PASSO
```

fig. 10

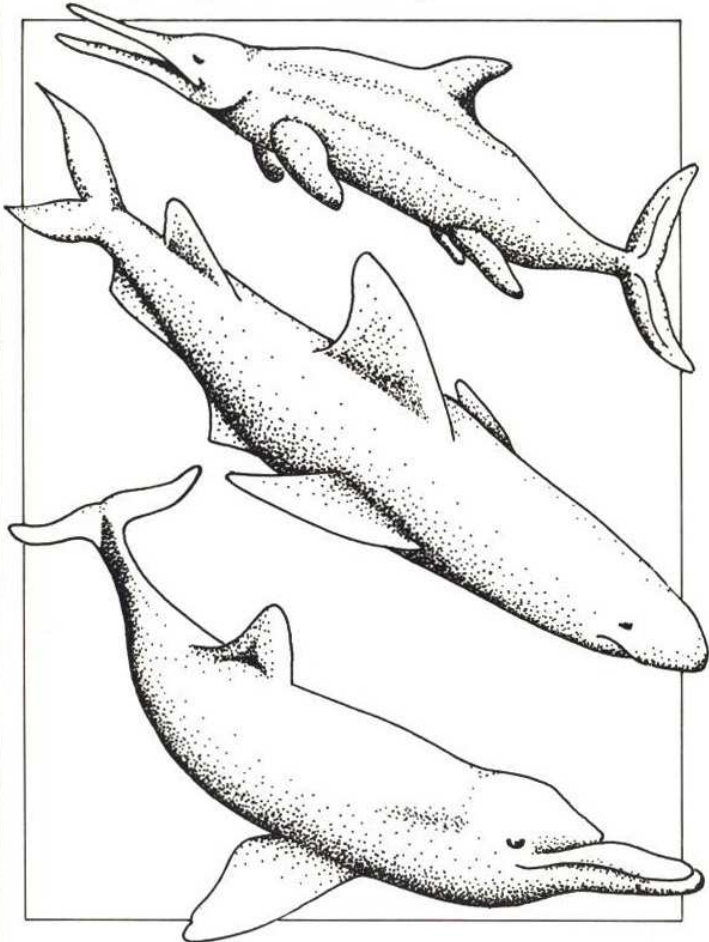
5º passo: dê agora uma recordada sobre o que você aprendeu a respeito de Darwin. Veja como a situação inicial pode evoluir dando origem a uma extinção total ou a "espécies" estáveis. Ocasionalmente obtemos como resultado final formas "meta-estáveis" que oscilam entre duas possibilidades.

Uma coisa muito interessante é se observar o desenrolar da "evolução convergente".

O tubarão, o golfinho, e o ictiossauro têm origens completamente diferentes: um é peixe, o segundo é mamífero e o terceiro é um réptil, (fig. 11). Ao se adaptarem, porém, ao meio aquático ficaram submetidos a conjuntos similares de regras e acabaram evoluindo para formas semelhantes.

Analogamente, as configurações iniciais, diferentes umas das outras pois são geradas ao ACASO, ficam submetidas a um conjunto de regras que simulam uma PRESSÃO DE EVOLUÇÃO e acabam convergindo para formas similares.

▷

A grande virtude deste programa é o de permitir simular, em poucos minutos, um suceder de gerações que dificilmente poderemos observar na prática. Para reiniciar a observação digite **BREAK** e a seguir **RUN.**

6º passo: para salvar o programa em fita você tem 3 caminhos:

a) Digitar **SAVE "LIFE"** normalmente

b) Acrescentar as linhas:

```
10 SAVE "LIFE"
11 RUN
```

e começar a gravação com

```
GOTO 10
```

c) Se você tiver um TK85, acrescente as linhas:

```
10 LET LIFE = USR 8405
11 RUN
```

Neste caso, ao digitar:

```
GOTO 10
```

você estará gravando o programa em HIGH SPEED.

Programa do Mês

ARQUIVO

Este programa foi adaptado a partir de um programa publicado em março na revista Microcomputing, sendo o original de autoria de Russel King.

É um programa inteiramente em BASIC que compensa sua lentidão com uma grande elasticidade. Como o programa é muito extenso (9800 bytes), aconselhamos o leitor a gravar a listagem em fita cada 50 ou 60 linhas: se faltar energia elétrica nem todo o trabalho de digitação estará perdido. Esta técnica tem a vantatem adicional de permitir uma "descansadinha" enquanto o programa incompleto está sendo gravado.

Quando o programa estiver completo, é conveniente gravá-lo algumas vezes (isto pode ser feito automaticamente: veja dicas na página 29).

Ao rodar o programa o leitor deve criar o hábito de digitar sempre **GOTO** 1 e **nunca RUN**, para não perder os dados a serem inseridos posteriormente. A primeira coisa que vemos é um menu inicial que caracteriza o computador empregado, dimensionando convenientemente a memória (fig. 1).

```
SEU COMPUTADOR E* :
   1...TK82-C  (16K)
   2...TK83    (16K)
   3...ZX81  (16K)
   4...TIMEX  (16K)
   5...CP 200
   6...NEZ8000(16K)
   7...TK82/83  (64K)
   8...TK85  (16K)
   9...TK85  (48K)


(MENU  INICIAL)
```

fig. 1

Uma vez digitada a opção correta, passamos ao menu principal (fig. 2). Quando usamos o programa pela primeira vez devemos começar pela opção 8:

8 . . . MONTAR NOVO FICHÁRIO

```
  COMANDOS  DO  ARQUIVO  MICROHOBBY
1..INSERIR  REGISTROS
2..TROCAR  REGISTROS
3..APAGAR  REGISTROS
4..LISTAR  TODOS  OS  REGISTROS
5..BUSCA  DE  UM  REGISTRO
6..ORDENAR  REGISTROS
7..GRAVAR  REGISTROS
8..MONTAR  NOVO  FICHARIO
9..LISTAR  CAMPOS

QUAL  VOCE  DESEJA  USAR?
```

fig. 2

O programa pergunta qual o nome do fichário: este nome se transforma também no nome do programa, quando usarmos a opção 7:

7 . . . GRAVAR REGISTROS

o argumento do **SAVE** será escolhido nesta ocasião.

Vamos supor que queiramos montar uma agenda de telefones de amigos. Digitamos, então:

AGENDA AMIGOS

O programa pergunta então quantos campos terá cada ficha. Como se trata de uma agenda de telefones bastam ▷

dois. Dependendo da utilização do fichário, uma quantidade maior de campos poderá ser escolhida. Obviamente quanto mais campos . . . menos fichas.

A seguir devemos definir quantos caracteres para o título de cada campo. Aqui convém ser bem econômico. No nosso exemplo vamos escolher três. Isto permite que o tamanho máximo de cada campo seja de 28 caracteres:

NOME DO CAMPO + CONTEÚDO DO CAMPO = 31 CARACTERES

O programa pede agora o nome do campo 1. Vamos digitar:

NOM (NOME DO AMIGO)

Devemos agora responder quantos caracteres o campo NOM deve ter. Como se trata de uma agenda de amigos, bastam uns 12 caracteres (não precisamos do nome completo).

Repetimos a operação para o campo 2:

Nome do campo 2: TEL (TELEFONE)
Comprimento: 7

Somos informados que a memória comporta 33Ø registros (com 16 K), e nos é perguntado quanto desejamos (fig. 3).

```
QUAL SERA" O NOME DO FICHARIO ?
FICHARIO : AGENDA AMIGOS
DESEJA QUANTOS CAMPOS?2
QUANTOS CARACTERES PARA O TITULO
DE CADA CAMPO ? (0 A 15)3
O TAMANHO MAXIMO DE CADA CAMPO
E" DE 28 CARACTERES

CAMPO 1:NOM
COMPRIMENTO DO CAMPO(1-28):12

CAMPO 2:TEL
COMPRIMENTO DO CAMPO(1-28):12

O NUMERO MAXIMO DE REGISTROS
VALE APROXIMADAMENTE 330
QUANTOS REGISTROS DESEJA?
█
```

fig. 3

Apenas para arredondar, vamos escrever 3ØØ. O computador pede então para digitar qualquer tela menos **BREAK**.

Voltamos então ao menu principal com o fichário já dimensionado. Esta volta não é muito rápida devido a uma peculiaridade do **SCROLL** do TK (veja DICAS, pág. 27).

Vamos agora inserir alguns registros. Para isto basta digitar 1:

1 . . . INSERIR REGISTROS

O computador, a cada inserção nos pergunta se desejamos trocar algo ou se desejamos inserir mais registros. Todas essas perguntas devem ser respondidas por S ou N (Sim ou Não).

Ao terminar a lista, digite N quando a pergunta for: "deseja inserir mais registros?". Não se preocupe se o computador parece sair do ar (mais uma vez esquisitices do **SCROLL**).

Digamos agora que você queira listar os registros. Basta digitar 4:

4.. LISTAR TODOS OS REGISTROS

O computador pergunta se queremos a listagem de todos os campos e se esta deve ser efetuada no vídeo ou na impressora.

A listagem surge na tela como **SCROLL** e a cada ficha pára. Para continuar basta ficar pressionando a tecla C. Querendo interromper digite **BREAK** e **GOTO** 1ØØØ (e tenha paciência com o **SCROLL**) (fig. 4).

```
QUER A LISTAGEM DE TODOS CAMPOS?
NO VIDEO OU NA IMPRESSORA? (V/I)
DIGITE C PARA CONTINUAR

NOM:JOSE ANTONIO
TEL:5438877

NOM:ALBERTO
TEL:2872456

NOM:VALERIA
TEL:4523387

NOM:VICENTE
TEL:4556734
```

fig. 4

Digamos que seja necessário tirar uma lista na impressora em ordem alfabética. Devemos, no menu principal digitar 6.

6 . . ORDENAR REGISTROS

Uma vez escolhido o campo a ser ordenado após a execução é perguntado se desejamos refazer a ordenação segundo outro critério. Digitando N, volta-se ao menu principal. Digitamos agora 4 e escolhemos como opção a impressora, obtendo, por exemplo, a listagem da fig. 5. ▷

```
AGENDA AMIGOS

NOM:ALBERTO
TEL:2872456

NOM:DELERMO
TEL:2114533

NOM:JOSE ANTONIO
TEL:5438677

NOM:MARCIA
TEL:8134555

NOM:MARILDA
TEL:8997656

NOM:VALERIA
TEL:4523387

NOM:VICENTE
TEL:4556734
```

fig. 5

Para trocar algum registro, basta digitar 2:

2 . . TROCAR REGISTROS

Sabendo o número da ficha a ser trocada podemos alterar qualquer um de seus campos (fig. 6):

```
ENTRE COM O NUMERO DO REGISTRO
A SER TROCADO

NOM:DELERMO
                IRA' TROCAR?
TEL:2114533
                IRA' TROCAR?
TEL:2114535

REGISTRO NUMERO 2 ACABADO
DESEJA TROCAR OUTRO REGISTRO?
```

fig. 6

Se não soubermos o número, entra em funcionamento a mesma rotina de busca do ítem 5:

5 . . BUSCAR DE UM REGISTRO

Digitando S para o campo NOM a busca começa. Ao invés de digitar o conteúdo completo do nome, vamos digitar apenas V (no fichário do nosso exemplo temos dois nomes que começam com V: Vicente e Valéria). O programa busca os dois (fig. 7).

```
PRESSIONE "S" QUANDO VOCE
TIVER O CAMPO
NOM ?
ENTRE COM O CONTEUDO DESTE CAMPO
REGISTRO NUMERO 6
NOM:VALERIA
TEL:4523387
DESEJA CONTINUAR A BUSCA?
REGISTRO NUMERO 7
NOM:VICENTE
TEL:4556734
DESEJA CONTINUAR A BUSCA?
BUSCA COMPLETA

DESEJA FAZER OUTRA BUSCA?
""
```

fig. 7

O campo 3:

3 . . APAGAR REGISTROS

tem funcionamento análogo. Por prudência, antes de apagar o programa, pede uma confirmação (fig. 8).

```
VOCE PRECISA ENTRAR COM O NUMERO
DO REGISTRO A SER APAGADO.
VOCE DESEJA FAZER UMA PESQUISA
SOBRE O NUMERO DOS REGISTROS?

ENTRE COM O NUMERO DO REGISTRO
A SER APAGADO
NOM:VICENTE
TEL:4556734
APAGAR ESTE REGISTRO?
""
```

fig. 8

O campo 9 mostra uma ficha vazia apenas para lembrar qual o formato do arquivo que está sendo rodado (fig. 9).

```
LISTAGEM DOS CAMPOS DE

[illegible]

NOME  :
END   :
CEP   :
TEL   :

PRESSIONE QUALQUER TECLA
MENOS "BREAK"
```

fig. 9

Finalmente, optando por 7 o leitor poderá gravar em fita os registros efetuados até então. Se o computador ▷

for um TK85 poderemos optar por velocidade normal ou **HIGH-SPEED.**

Lembre-se, porém, de uma coisa importantíssima. Se o programa for interrompido por algum erro jamais digite **RUN**, sob pena de perder todo o fichário. O programa deve ser sempre reiniciado com um **GOTO** 1000.

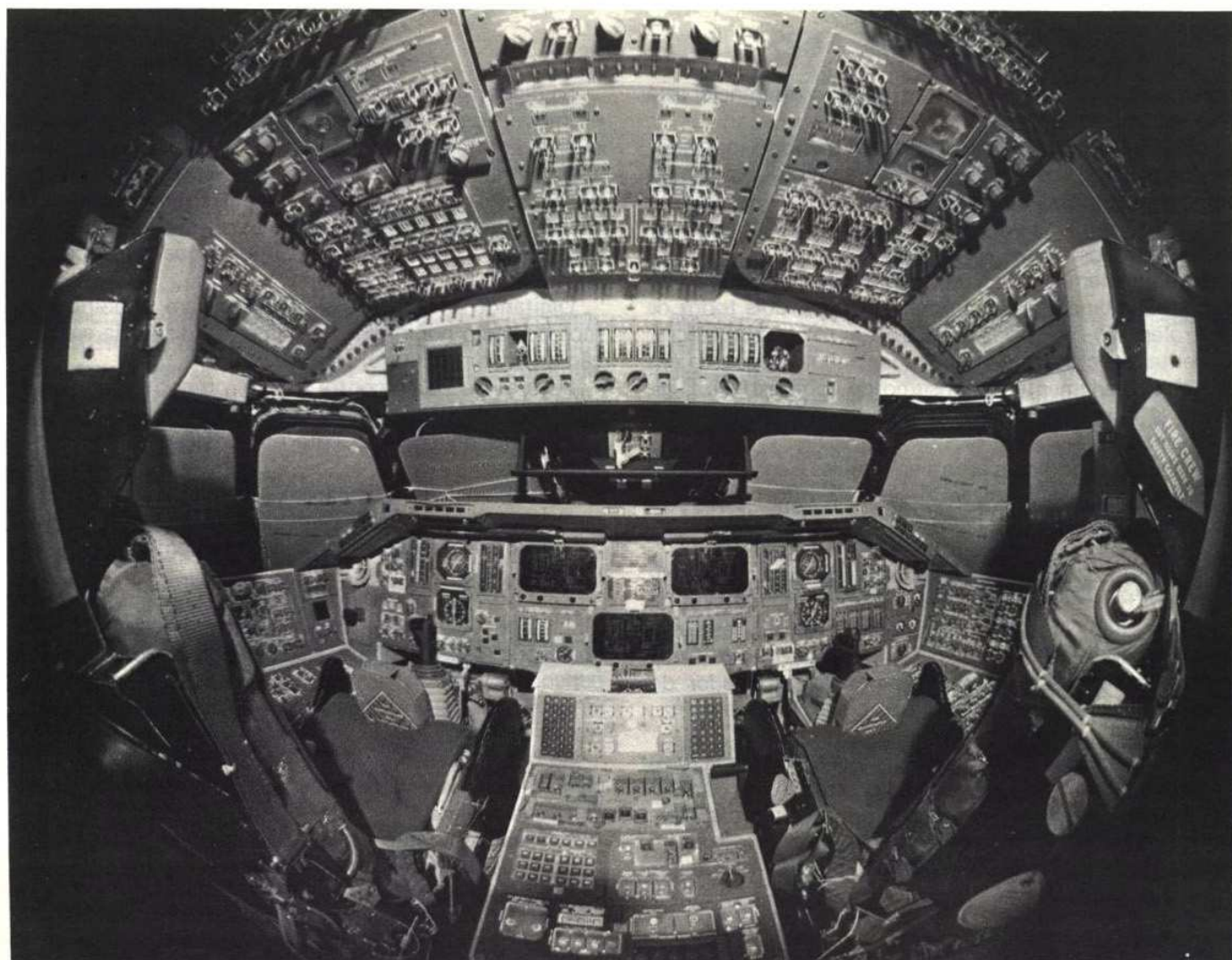
```
   0 REM MICROMERA - 1983
 900 REM ESCOLHA DO COMPUTADOR
 901 PRINT "SEU COMPUTADOR E":";
TAB 2;"1...TK82-C (16K)";TAB 2;"
2...TK83    (16K)";TAB 2;"3...ZX8
1 (16K)";TAB 2;"4...TIMEX (16K)"
;TAB 2;"5...CP 200";TAB 2;"6...N
EZ8000(16K)";TAB 2;"7...TK82/83
(64K)";TAB 2;"8...TK85 (16K)";TA
B 2;"9...TK85 (48K)"
 910 LET HI=0
 915 INPUT A
 920 LET A=INT A
 930 IF A<1 OR A>9 THEN GOSUB 60
00
 940 IF A>0 AND A<10 THEN GOTO 9
55
 950 GOTO 900
 955 LET MEMO=39000
 960 IF A<>7 AND A<>9 THEN LET M
EMO=6300
 970 IF A=8 OR A=9 THEN LET HI=1
1000 REM  MENU
1005 FAST
1010 CLS
1030 PRINT TAB 2;"COMANDOS DO AR
QUIVO MICROHOBBY"
1040 PRINT
1050 PRINT "1..INSERIR REGISTROS
"
1060 PRINT
1070 PRINT "2..TROCAR REGISTROS"
1080 PRINT
1090 PRINT "3..APAGAR REGISTROS"
1100 PRINT
1110 PRINT "4..LISTAR TODOS OS R
EGISTROS"
1120 PRINT
1130 PRINT "5..BUSCA DE UM REGIS
TRO"
1140 PRINT
1150 PRINT "6..ORDENAR REGISTROS
"
1160 PRINT
1170 PRINT "7..GRAVAR REGISTROS"
1180 PRINT
1190 PRINT "8..MONTAR NOVO FICHA
RIO"
1200 PRINT
1210 PRINT "9..LISTAR CAMPOS"
1220 PRINT
1230 PRINT
1240 PRINT "QUAL VOCE DESEJA USA
R?";
1250 INPUT A
1260 LET A=INT A
1270 IF A<1 OR A>9 THEN GOSUB 60
00
1275 IF A>0 AND A<10 THEN GOTO 1
310
1280 CLS
1300 GOTO 1030
1310 CLS
1320 IF A=1 THEN GOSUB 1420
1330 IF A=2 THEN GOSUB 1710
1340 IF A=3 THEN GOSUB 2170
1350 IF A=4 THEN GOSUB 2670
1360 IF A=5 THEN GOSUB 3300
1370 IF A=6 THEN GOSUB 3830
1380 IF A=7 THEN GOSUB 4132
1390 IF A=8 THEN GOSUB 4250
1400 IF A=9 THEN GOSUB 4920
1405 FAST
1410 GOTO 1010
1420 REM INSERIR REGISTROS
1430 LET N=N+1
1440 SCROLL
1450 PRINT "REGISTRO NUMERO ";N
1460 IF N<=M THEN GOTO 1510
1470 LET N=M
1480 SCROLL
1482 SLOW
1484 FOR I=1 TO 12
1490 PRINT AT 11,2;"FIM DOS REGI
STROS"
1491 PRINT AT 11,2;"FIM DOS REGI
STROS"
1494 NEXT I
1496 FAST
1500 GOTO 1700
1510 FOR I=1 TO N1
1520 SCROLL
1530 PRINT N$(I);":";
1540 INPUT I$(N,D(I,1) TO D(I,2)
)
1550 PRINT I$(N,D(I,1) TO D(I,2)
)
1560 SCROLL
1570 NEXT I
1580 SCROLL
1590 PRINT "QUER TROCAR ALGUMA C
OISA?";
1600 INPUT A$
1610 SCROLL
1620 IF A$="S" THEN GOTO 1510
1630 SCROLL
1640 PRINT "REGISTRO ";N;" INSER
IDO"
1650 SCROLL
1660 PRINT "DESEJA INSERIR MAIS
REGISTROS?";
1670 INPUT A$
1680 SCROLL
1690 IF A$="S" THEN GOTO 1420
1700 RETURN
1710 REM TROCAR REGISTRO
1720 SCROLL
1730 SCROLL
1740 PRINT "PARA TROCAR UM REGIS
TRO VOCE"
1750 SCROLL
1760 PRINT "PRECISA ENTRAR COM O
 SEU NUMERO"
1770 SCROLL
1780 PRINT " VOCE DESEJA PESQUIS
AR SOBRE"
1790 SCROLL
1800 PRINT "O NUMERO DO REGISTRO
?"
1810 INPUT A$
1820 SCROLL
1830 IF A$="S" THEN GOSUB 3300
1840 CLS
1850 SCROLL
1860 PRINT "ENTRE COM O NUMERO D
O REGISTRO"
1870 SCROLL
1880 PRINT "A SER TROCADO"
1890 SCROLL
1900 INPUT A
1910 LET A=INT A
1920 SCROLL
1930 IF A>0 AND A<=N THEN GOTO 1
960
1940 PRINT "NUMERO DE REGISTRO N
AO VALIDO"
1950 GOTO 2110
1960 FOR I=1 TO N1
1970 SCROLL
1980 PRINT N$(I);":";I$(A,D(I,1)
 TO D(I,2))
1990 SCROLL
```

```
2000 PRINT TAB 10;"IRA" TROCAR?"
;
2010 INPUT A$
2020 SCROLL
2030 IF A$<>"S" THEN GOTO 2080
2040 PRINT N$(I);":";
2050 INPUT I$(A,D(I,1) TO D(I,2)
)
2060 PRINT I$(A,D(I,1) TO D(I,2)
)
2070 SCROLL
2080 NEXT I
2090 SCROLL
2100 PRINT "REGISTRO NUMERO ";A;
" ACABADO"
2110 SCROLL
2120 PRINT "DESEJA TROCAR OUTRO
REGISTRO?"
2130 INPUT A$
2140 SCROLL
2150 IF A$="S" THEN GOTO 1710
2160 RETURN
2170 REM APAGAR UM REGISTRO
2180 IF N>0 THEN GOTO 2220
2190 SCROLL
2200 PRINT "NENHUM REGISTRO NO F
ICHARIO"
2210 GOTO 2510
2220 SCROLL
2230 PRINT "VOCE PRECISA ENTRAR
COM O NUMERO"
2240 SCROLL
2250 PRINT "DO REGISTRO A SER AP
AGADO."
2260 SCROLL
2270 PRINT "VOCE DESEJA FAZER UM
A PESQUISA"
2280 SCROLL
2290 PRINT "SOBRE O NUMERO DOS R
EGISTROS ?"
2300 INPUT A$
2310 SCROLL
2320 IF A$="S" THEN GOSUB 3300
2330 SCROLL
2340 PRINT "ENTRE COM O NUMERO D
O REGISTRO "
2350 SCROLL
2360 PRINT "A SER APAGADO"
2370 INPUT A
2380 LET A=INT A
2390 IF A>0 AND A<=N THEN GOTO 2
430
2400 SCROLL
2410 PRINT "NUMERO DE REGISTRO N
AO VALIDO"
2420 GOTO 2610
2430 FOR I=1 TO N1
2440 SCROLL
2450 PRINT N$(I);":";I$(A,D(I,1)
 TO D(I,2))
2460 NEXT I
2470 SCROLL
2480 PRINT "APAGAR ESTE REGISTRO
?"
2490 INPUT A$
2500 IF A$="S" THEN GOTO 2520
2510 GOTO 2610
2520 IF A=N THEN GOTO 2580
2540 FOR I=A TO N-1
2550 LET I$(I)=I$(I+1)
2560 NEXT I
2580 LET N=N-1
2590 SCROLL
2600 PRINT "REGISTRO APAGADO"
2610 SCROLL
2620 PRINT "MAIS ALGUM REGISTRO
PARA APAGAR?"
2630 INPUT A$
2640 SCROLL
2650 IF A$="S" THEN GOTO 2180
2660 RETURN
2670 REM LISTAGEM DOS REGISTROS
2680 SCROLL
2690 LET A=0
2700 PRINT "QUER A LISTAGEM DE T
ODOS CAMPOS?"
2710 INPUT A$
2720 IF A$="S" THEN LET A=1
2730 SCROLL
2750 FOR I=1 TO N1
2760 LET A(I)=A
2770 IF A=1 THEN LET A(I)=I
2780 NEXT I
2800 LET J=N1
2810 IF A=1 THEN GOTO 2970
2820 SCROLL
2830 PRINT "PRESSIONE ""S"" PARA
 OS CAMPOS"
2840 SCROLL
2850 PRINT "QUE VOCE DESEJA LIST
AR"
2860 LET J=0
2870 SCROLL
2880 FOR I=1 TO N1
2890 SCROLL
2900 PRINT N$(I);" ?";
2910 INPUT A$
2920 PRINT A$
2930 IF A$<>"S" THEN GOTO 2960
2940 LET J=J+1
2950 LET A(J)=I
2960 NEXT I
2970 SCROLL
2980 PRINT "NO VIDEO OU NA IMPRE
SSORA? (V/I)"
2990 INPUT A$
3000 IF A$(1)="I" THEN GOTO 3130
3005 SLOW
3010 SCROLL
3015 PRINT "DIGITE C PARA CONTIN
UAR"
3020 SCROLL
3030 PRINT TAB 5;L$
3040 SCROLL
3050 FOR I=1 TO N
3060 FOR K=1 TO J
3070 SCROLL
3080 PRINT N$(A(K));":";I$(I,D(A
(K),1) TO D(A(K),2))
3090 NEXT K
3100 SCROLL
3106 IF INKEY$<>"C" THEN GOTO 31
05
3110 NEXT I
3115 FAST
3120 GOTO 3250
3140 CLS
3150 LPRINT TAB 5;L$
3155 LPRINT
3160 LPRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒"
3170 FOR I=1 TO N
3180 FOR K=1 TO J
3190 LPRINT
3200 LPRINT N$(A(K));":";I$(I,D(
A(K),1) TO D(A(K),2))
3210 NEXT K
3220 LPRINT
3225 LPRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒"
3230 NEXT I
3250 SCROLL
3260 PRINT "OUTRA LISTAGEM?";
3270 INPUT A$
```

▷

```
3280 IF A$="S" THEN GOTO 2670
3290 RETURN
3300 REM BUSCA DE UM REGISTRO
3310 SCROLL
3320 PRINT "PRESSIONE ""S"" QUAN
DO VOCE "
3330 SCROLL
3340 PRINT "TIVER O CAMPO"
3350 FOR I=1 TO N1
3360 SCROLL
3370 PRINT N$(I);" ?";
3380 INPUT A$
3390 IF A$="S" THEN GOTO 3440
3400 NEXT I
3410 SCROLL
3420 PRINT "SELECAO CANCELADA"
3430 GOTO 3780
3440 SCROLL
3450 PRINT "ENTRE COM O CONTEUDO
 DESTE CAMPO"
3460 INPUT A$
3470 LET A=LEN A$
3480 IF A<1 THEN GOTO 3450
3490 IF A<=(D(I,2)-D(I,1)+1) THE
N GOTO 3520
3500 LET A=(D(I,2)-D(I,1)+1)
3510 LET A$=A$(1 TO A)
3520 LET A=A-1+D(I,1)
3530 LET F=0
3540 LET Q=0
3560 FOR J=1 TO N
3570 IF A$=I$(J,D(I,1) TO A) THE
N GOSUB 3660
3580 IF Q=1 THEN GOTO 3600
3590 NEXT J
3610 SCROLL
3620 PRINT "BUSCA COMPLETA"
3630 SCROLL
3640 IF F=0 THEN PRINT "REGISTRO
 NAO ACHADO"
3650 GOTO 3780
3660 SCROLL
3670 PRINT "REGISTRO NUMERO ";J
3680 FOR K=1 TO N1
3690 SCROLL
3700 PRINT N$(K);":";I$(J,D(K,1)
 TO D(K,2))
3710 NEXT K
3720 SCROLL
3730 PRINT "DESEJA CONTINUAR A B
USCA?"
3740 INPUT B$
3750 IF B$="N" THEN LET Q=1
3760 LET F=1
3770 RETURN
3780 SCROLL
3790 PRINT "DESEJA FAZER OUTRA B
USCA?"
3800 INPUT A$
3810 IF A$="S" THEN GOTO 3300
3820 RETURN
3830 REM ORDEM DOS REGISTROS
3840 SCROLL
3850 PRINT "ESTA ROTINA ORDENA A
LFANUMERICA-"
3860 SCROLL
3870 PRINT "MENTE COM BASE NUM C
AMPO"
3880 SCROLL
3890 PRINT "PRESSIONE ""S"" PARA
 O CAMPO "
3900 SCROLL
3910 PRINT "QUE IRA SERVIR DE B
ASE"
3920 SCROLL
3930 FOR I=1 TO N1
3940 SCROLL
3950 PRINT N$(I);" ?"
3960 INPUT A$
3970 IF A$="S" THEN GOTO 4020
3980 NEXT I
3990 SCROLL
4000 PRINT "ORDENACAO CANCELADA"
4010 GOTO 4160
4020 SCROLL
4040 FOR J=1 TO N-1
4050 FOR K=J TO N
4060 IF I$(J,D(I,1) TO D(I,2))<=
I$(K,D(I,1) TO D(I,2)) THEN GOTO
 4073
4070 LET B$=I$(J)
4071 LET I$(J)=I$(K)
4072 LET I$(K)=B$
4073 NEXT K
4074 NEXT J
4076 SCROLL
4077 PRINT "ORDENACAO COMPLETA"
4080 SCROLL
4090 PRINT "ORDENAR COM BASE EM
OUTRO CAMPO?"
4100 INPUT A$
4110 SCROLL
4120 IF A$="S" THEN GOTO 3830
4130 RETURN
4132 REM GRAVAR PROGRAMA
4134 IF HI=0 THEN GOTO 4146
4135 PRINT "PREFERE GRAVAR EM VE
LOCIDADE    NORMAL (N) OU HIGH-S
PEED?(H)"
4137 INPUT C$
```

▷

```
4138 IF C$="N" THEN GOTO 4146
4139 CLS
4140 PRINT "COM O GRAVADOR EM PO
SICAO DE     GRAVAR DIGITE A TECL
A H"
4141 INPUT A$
4142 IF A$="H" THEN LET H=USR 84
05
4145 GOTO 1000
4146 CLS
4150 SCROLL
4160 PRINT "PONHA O GRAVADOR EM
POSICAO DE"
4170 SCROLL
4180 PRINT "GRAVAR. PARA COMECAR
 A GRAVAR"
4190 SCROLL
4200 PRINT "PRESSIONE QUALQUER T
ECLA"
4210 PAUSE 3000
4220 SCROLL
4230 SAVE L$
4240 GOTO 1000
4250 REM MONTAR FICHARIO
4260 SCROLL
4270 PRINT "QUAL SERA" O NOME DO
 FICHARIO ?"
4272 INPUT Q$
4273 DIM L$(LEN Q$)
4274 FOR Q=1 TO LEN Q$
4276 LET L$(Q)=CHR$ (CODE Q$(Q)+
128)
4278 NEXT Q
4290 SCROLL
4292 PRINT "FICHARIO : ";L$
4294 SCROLL
4300 PRINT "DESEJA QUANTOS CAMPO
S?";
4320 INPUT N1
4330 PRINT N1
4340 SCROLL
4350 LET K=15
4360 PRINT "QUANTOS CARACTERES P
ARA O TITULO"
4370 SCROLL
4380 PRINT "DE CADA CAMPO ? (0 A
 15)";
4390 INPUT K
4400 IF K>15 THEN LET K=15
4410 IF K<0 THEN LET K=0
4415 PRINT K
4420 SCROLL
4430 LET L=31-K
4440 PRINT "O TAMANHO MAXIMO DE
CADA CAMPO"
4450 SCROLL
4460 PRINT "E" DE ";L;" CARACTER
ES"
4470 DIM N$(N1,K)
4480 DIM A(N1)
4490 DIM D(N1,2)
4500 SCROLL
4510 SCROLL
4520 PRINT "ENTRE COM DESCRICOES
 DO CAMPO"
4530 SCROLL
4540 PRINT "E O COMPRIMENTO DO C
AMPO"
4550 LET J=1
4560 SCROLL
4570 FOR I=1 TO N1
4580 SCROLL
4590 LET A$=""
4600 PRINT "CAMPO ";I;":";
4610 INPUT A$
4620 LET A$=A$+"
"
4630 LET N$(I)=A$
4635 PRINT N$(I)
4640 SCROLL
4650 PRINT "COMPRIMENTO DO CAMPO
(1-";L;"):";
4660 INPUT A
4670 IF A<1 THEN LET A=1
4680 IF A>L THEN LET A=L
4690 PRINT A
4700 LET D(I,1)=J
4710 LET J=J+A
4720 LET D(I,2)=J-1
4730 SCROLL
4740 NEXT I
4750 LET J=J-1
4760 LET N=0
4770 LET M=INT ((MEMO-N1*(31-L+1
2))/J)
4780 SCROLL
4790 PRINT "O NUMERO MAXIMO DE R
EGISTROS "
4800 SCROLL
4810 PRINT "VALE APROXIMADAMENTE
 ";M
4820 SCROLL
4830 PRINT "QUANTOS REGISTROS VO
CE DESEJA?"
4840 INPUT A
4850 LET A=INT A
4860 IF A>0 AND A<M THEN LET M=A
4870 DIM I$(M,J)
4872 SCROLL
4874 PRINT M
4880 SCROLL
4882 FAST
4890 PRINT "DIGITE QQUER TECLA(M
ENOS BREAK"
4900 PAUSE 30000
4910 RETURN
4920 REM LISTAGEM DOS CAMPOS
4930 SCROLL
4940 PRINT "LISTAGEM DOS CAMPOS
DE"
4945 SCROLL
4946 SCROLL
4947 PRINT L$
4950 SCROLL
4952 SCROLL
4954 PRINT "██████████████████████
██"
4960 FOR I=1 TO N1
4980 SCROLL
4990 PRINT N$(I);":"
5000 NEXT I
5010 SCROLL
5012 SCROLL
5014 PRINT "██████████████████████
██████████"
5016 SCROLL
5020 PRINT "PRESSIONE QUALQUER T
ECLA"
5021 SCROLL
5022 PRINT "MENOS ""BREAK"""
5030 SCROLL
5040 PAUSE 30000
5050 RETURN
5060 STOP
6000 SLOW
6010 FOR I=1 TO 12
6030 PRINT AT 0,0;"POR FAVOR ENT
RE █ E █"
6040 PRINT AT 0,0;"POR FAVOR ENT
RE 1 E 9"
6050 NEXT I
6060 FAST
6070 RETURN
7000 STOP
```

A FITA DO MÊS

Simulador de Vôo

A fita que estamos oferecendo de brinde para os nossos assinantes contém um jogo bastante realista de 16 K intitulado "SIMULADOR DE VÔO". A adaptação feita para o Brasil de um programa originalmente inglês criou um pequeno grilo, felizmente já sanado, para o TK85 (veja DICAS do nº 2).

Inicialmente o programa pergunta se queremos o vôo completo ou apenas a fase de aterrisagem. Aconselhamos os "manicacas" a se limitarem inicialmente à aterrisagem. Por maior motivo, ao serem questionados se desejam ou não os efeitos do vento, aconselhamos responder "Não".

Ao começar o programa, inicialmente temos, uma visão do interior da cabine. Pelo parabrisa vemos a linha do horizonte. Puxando o manche para trás (com joystick, ou com a tecla 6) vemos o horizonte descer, pois o nariz do avião está apontando para cima.

Da mesma forma, digitando 7 fazemos o nariz baixar e vemos o horizonte subindo. O nível do horizonte está indicado num mostrador à esquerda.

Cuidado, pois este instrumento tem um pouco de inércia. Usando as teclas 5 e 7 fazemos o avião se inclinar para esquerda e para a direita. Obviamente veremos o ▷

horizonte se inclinar ao contrário, pois o referencial está fixo na cabine e não na Terra.

Abaixo do mostrador de horizonte temos o indicador de combustível, cuja indicação não é crítica quando escolhemos apenas a fase de aterrisagem, mas que se torna um grande fator de angústia quando o vôo é longo e estamos perdidos.

À direita temos o indicador do regime de rotação do motor. Digitando P (de forma contínua, pois todos os controles são do tipo **INKEY$**) a rotação aumenta, aumentando a velocidade indicada logo à esquerda em KM/H e, obviamente, o consumo de combustível. Digitando O o regime diminui: (cuidado para não perder a sustentação).

O altímetro indica a altura em metros e sua marcação pode ser alterada levantando ou baixando o nariz do avião. Cuidado que, ao perder altura, o avião ganha velocidade.

Digitando F o flap é abaixado, aumentando a sustentação em baixa velocidade (aterrisagem) e D ele é retirado. Dar flap em alta velocidade produz desastres.

Para abaixar ou levantar o trem de aterrisagem basta digitar G. Antes de aterrisar verifique o mostrador do trem ao lado do de flap, para não correr o risco de raspar a barriga na pista.

No lugar de honra do painel temos a bússola que mostra o rumo do avião. À direita da bússola temos o rádiogoniômetro que indica o rumo (em graus) e a distância (em KMS) até o radiofarol do aeroporto. Este rumo é mostrado também de forma analógica pelo mostrador circular com modelo de avião.

Se digitarmos M (mapa), a cabine desaparece do vídeo e em seu lugar vemos o mapa da região. No alto à esquerda continuamos tendo acesso à leitura do rumo do avião. O avião e o rádio-farol são representados como pontos piscando. Querendo fixar o sistema de navegação em um rádio-farol situado em outra cidade basta digitar B até que a cidade escolhida fique piscando. Em vôos longos isto pode ajudar.

Digitando V novamente voltamos para a cabine e vemos no rádio goniômetro o rumo e a distância em relação ao novo rádio-farol fixado.

Se o avião bater contra o solo, ou contra os morros indicados no mapa, haverá um quebra-quebra e veremos o relatório do acidente para nos orientar nas próximas tentativas.

A fase final da aterrisagem é a mais emocionante e lembra os velhos tempos em que a maior preocupação do piloto era de não deixar o cachecol embaraçar na hélice.

Com o vídeo mostrando a cabine, digitar V: se a distância até o aeroporto for suficientemente pequena, ele aparecerá em perspectiva e a habilidade do piloto não dependerá mais tanto dos instrumentos. Apesar disto, ainda são visíveis o centralizador de descida, o velocímetro e o altímetro.

Se nesse momento você lembrar do trem de aterrisagem, é só digitar V novamente que o painel da cabine volta a aparecer.

Em pouco tempo você se tornará um hábil piloto, tanto visual quando em vôo por instrumentos.

Dicas

Apagando a Tela

Quando montamos uma tela com muitas linhas usando o **SCROLL**, nos defrontamos normalmente com um problema: apagar a tela com um **CLS** é uma tarefa que pode ser muito demorada. Quem já brincou com o simulador de vôo já sentiu isso na mensagem final: DIGITE **RUN** PARA VOAR NOVAMENTE. Digitando **RUN** (e **NEW LINE**) deve se esperar um bocado para que a tela se apague totalmente e o jogo reinicie.

Vamos então fazer algumas experiências:

Digite inicialmente o programa listado na fig. 1 (**A$** tem 32 o's). Com o computador no modo **SLOW** rode o programa.

```
110 LET A$="OOOOOOOOOOOOOOOOOOO
OOOOOOOOOOOOO"
120 FOR I=0 TO 6
130 SCROLL
140 SCROLL
150 SCROLL
160 PRINT A$
170 NEXT I
190 CLS
200 GOTO 100
```

fig. 1

Note quanto tempo demora o **CLS** para apagar a tela.

Tente dar um **BREAK** enquanto a tela está sendo apagada: inútil. É necessário esperar que a tarefa se complete.

Como evitar isso? Uma solução é colocar o computador em **SLOW** enquanto a tela está sendo escrita, mas em **FAST** quando está sendo apagada.

Acrescente as linhas:

```
100 SLOW
```

e

```
180 FAST
```

ao seu programa, de maneira que ele fique como na fig. 2.

```
100 SLOW
110 LET A$="OOOOOOOOOOOOOOOOOOO
OOOOOOOOOOOOO"
120 FOR I=0 TO 6
130 SCROLL
140 SCROLL
150 SCROLL
160 PRINT A$
170 NEXT I
180 FAST
190 CLS
200 GOTO 100
```

fig. 2

Rode o programa assim alterado e veja como as coisas melhoram um pouco.

Mas, será que não existe uma maneira mais elegante?

Podemos utilizar uma subrotina em linguagem de máquina, para encher a tela de espaços vazios.

Uma subrotina em linguagem de máquina é uma sequência de bytes que deve ser inseridas em algum lugar da memória e é chamada pela função **USR**. Toda subrotina desse tipo tem uma última instrução análoga ao **RETURN**, que faz voltar ao programa em BASIC. No curso de ASSEMBLY que estamos publicando discutem-se as melhores localizações para estas subrotinas. Por enquanto vamos localizá-la numa linha **REM** do programa em BASIC.

Digite então:

1 **REM** . . .   (3 pontos)

Sem apagar o programinha anterior acrescente as linhas:

```
10 FOR I = 16509 TO 16517
20 PRINT I, PEEK I
30 NEXT I
40 STOP
```

e agora digite **RUN**.

A função **PEEK** lê o byte contido no endereço de seu argumento. Desta forma o programinha em questão está lendo os bytes contidos nos endereços de 16509 a 16517.

O endereço 16509, no TK, contém o primeiro byte do programa em BASIC. Pegue o manual do TK (Apêndice A) e veja o que podemos deduzir do que mostra a tela (fig. 3).

```
16509          0
16510          1
16511          5
16512          0
16513          234
16514          27
16515          27
16516          27
16517          118
```

fig. 3

Os endereços 16509 e 16510 contém os bytes que indicam o número da linha. No nosso caso trata-se da linha 01.

Os endereços 16512 e 16511 (nesta ordem) mostram o comprimento da linha (no nosso caso 05 bytes, incluindo um byte final que indica que a linha acabou.

O endereço 16513 contém o byte 234. No apêndice A vemos que ele corresponde ao **REM**. A seguir temos ▷

os 3 pontos 27, 27 e 27 e finalmente o byte que indica **NEW LINE**, ou seja, esta linha de programa acabou (fig. 4).

ENDEREÇO	BYTE	SIGNIFICADO
16509	Ø	linha 1
16510	1	
16511	5	comprimento
16512	Ø	5 bytes
16513	234	**REM**
16514	27	.
16515	27	.
16516	27	.
16517	118	**NEW LINE**

fig. 4

Como podemos observar, o primeiro byte livre após o **REM** é o de endereço 16514, que por enquanto, contém o byte 27 (ponto).

Se quisermos alterar seu conteúdo basta usar a instrução **POKE**:

DIGITE **POKE** 16514, 2Ø1

e liste o programa novamente.

Olhe para a linha 1 (fig. 5).

```
   1 REM TAN ..
  10 FOR I=16509 TO 16517
  20 PRINT I,PEEK I
  30 NEXT I
  40 STOP
 100 SLOW
 110 LET A$="00000000000000000000
000000000000"
 120 FOR I=0 TO 6
 130 SCROLL
 140 SCROLL
 150 SCROLL
 160 PRINT A$
 170 NEXT I
 180 FAST
 190 CLS
 200 GOTO 100
```

fig. 5

O primeiro dos três pontos, justamente o que estava no endereço 16514, foi substituido por **TAN**, que é justamente o caracter 2Ø1. Rode o programa e você verá esta substituição (fig. 6).

```
16509          0
16510          1
16511          5
16512          0
16513          234
16514          201
16515          27
16516          27
16517          118
```

fig. 6

Vamos agora apagar as linhas 1, 1Ø, 2Ø, 3Ø, 4Ø, 1ØØ e 18Ø, voltando, por tanto, ao programa da figura 1.

Como a subrotina que vamos introduzir tem 21 bytes, vamos digitar uma linha 1 **REM** seguida de 21 pontos para reservar 21 endereços na memória.

A seguir, para não ter que fazer 21 **POKE's** seguidos vamos digitar esta pequena rotina:

1Ø **FOR** I = 16514 **TO** 16514 + 2Ø
2Ø **PRINT** I,
3Ø **INPUT** P
4Ø **PRINT** P
5Ø **POKE** I, P
6Ø **NEXT** I
7Ø **STOP**

e rode-a com o computador em **SLOW**. Entre com os bytes listados na figura 7. Liste o programa e veja como ficou estranha a linha 1 **REM** (fig. 8).

```
16514          42
16515          12
16516          64
16517          1
16518          213
16519          2
16520          35
16521          126
16522          254
16523          118
16524          40
16525          3
16526          62
16527          0
16528          119
16529          11
16530          120
16531          177
16532          200
16533          24
16534          241
```

fig. 7

```
   1 REM E£RND■STR$ ■7 ?"?■COS
 LET
  10 FOR I=16514 TO 16514+20
  20 PRINT I,
  30 INPUT P
  40 PRTNT P
```

fig. 8

Agora você já tem uma pequena subrotina em linguagem de máquina na linha **REM**. Apague as linhas 1Ø, 2Ø, 3Ø, 4Ø, 5Ø, 6Ø e 7Ø, digite:

1ØØ **SLOW**

e rode o programa. Tudo continua como no começo, pois o programa em BASIC ignora tudo o que estiver numa linha **REM**. Seu programa em linguagem de máquina está bem quietinho e só será despertado por um **USR**.

Vamos agora chamar a subrotina. A linha :

19Ø **CLS**

pode ser substituída por

19Ø **LET** LIMPEZA = **USR** 16514
195 **CLEAR**

A variável LIMPEZA não serve para nada: sua única finalidade é poder fazer constar da linha **USR** 16514. A linha não poderia começar por **USR** pois esta é uma função e não uma KEY-WORD. O argumento 16514 corresponde ao primeiro byte após o **REM** da linha 1.

Rode agora o programa assim alterado. Bem mais rápido não?

Dê um **BREAK** enquanto a tela estiver enchendo. Se a seguir você digitar **NEW LINE**, deverá outra vez esperar que a tela se apague lentamente.

Vamos agora fazer outra experiência: no endereço 16527 foi colocado o byte do caractere que precisa encher a tela. No nosso caso, o byte era Ø, correspondente a um espaço vazio.

Vamos escolher um outro caractere correspondente ao byte 146.

Vamos digitar então:

POKE 16527,146

Digite **RUN** e veja o que acontece.

Dê um **BREAK** e liste o programa.

Vamos produzir agora uma última loucura (esta dica vai ter um final apoteótico).

Digite esta rotina:

```
5 SLOW
1Ø FOR C = 128 TO 191
2Ø POKE 16527, C
3Ø RAND USR 16514
4Ø NEXT C
5Ø STOP
```

e rode o programa com **RUN**.

Você saberia explicar o que aconteceu?

Dicas

Gravação automática

Um dos primeiros erros cometidos pelos principiantes em TK é o de comprar fitas cassete de longa duração e nelas gravar uma grande quantidade de programas. Após um certo período de experiência percebe-se que esta técnica apresenta dois problemas:

1) Fitas de longa duração são mais delgadas e mais frágeis.
2) Muitos programas numa única fita produzem buscas demasiadamente demoradas.

O aconselhável, então, é comprar fitas de curta duração e gravar poucos programas nelas.

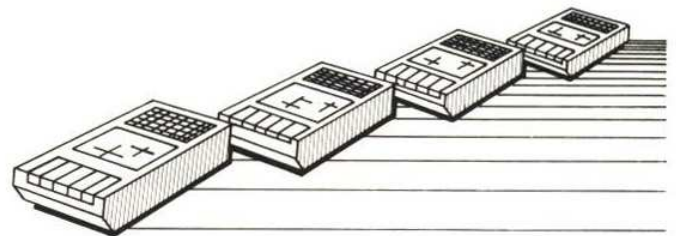
O ideal seria a gravação de um único programa por fita. A busca seria imediata.

Isto pode ser feito quando dispomos de programas longos.

Se os programas forem curtos (1 a 4 K) e não exigirem armazenamento de dados, podemos digitá-los todos numa única sequência colocando no programa um menú inicial proporcionando a escolha.

Obviamente este processo aumenta um pouco o tempo de transferência de fita para a RAM do computador e exige expansão de memória.

O tempo gasto na transferência, porém, é altamente compensado pela economia na busca do programa.

Entretanto, por mais curta que seja a fita, dificilmente ela será totalmente tomada pelo programa. Vale a pena, então, gravar o mesmo programa repetidas vezes: um pequeno acidente de gravação, uma interferência de uma emissora de FM, uma falha no revestimento da fita, põe o programa a perder. Uma gravação múltipla já diminui o risco de perda (lembre-se da Lei de Murphy).

Para que o processo de repetição da gravação não se torne enfadonho, podemos usar um pequeno truque.

Digamos, por exemplo, que se queira gravar, várias vezes, o programa "ARQUIVO" que é oferecido aos leitores neste número. Basta então, acrescentar ao fim do programa as linhas mostradas na fig. 1, colocar a fita no começo, dar o comando

GOTO 9ØØØ

e ir tomar um cafezinho.

```
9000 SAVE "ARQUIVO"
9010 PRINT "DIGITE ""BREAK"""
9020 PAUSE 300
9030 GOTO 9000
```

fig. 1

A linha 9Ø3Ø fará o processo se repetir **n** vezes.

Quando este programa for carregado, novamente no computador, aparecerá a mensagem:

DIGITE "**BREAK**"

mas não se apresse; se a pausa de tempo for superada o programa põe o computador no modo "**SAVE**", mas não é apagado da memória. Os riscos que aparecem na tela, inclusive, são característicos da emissão e não da recepção. Basta digitar "**BREAK**" a qualquer momento e **NEW LINE**, que veremos a listagem do programa.

Vamos então organizar nossa biblioteca de programas e cuidado para não tomar cafezinhos em excesso.

PROGRAMA DO LEITOR

Crepe

Luiz G.C. Antunes

Este jogo é um jogo de dados muito conhecido nos cassinos de todo o mundo.

Joga-se com fichas, voce joga contra a banca, que, neste caso é o computador.

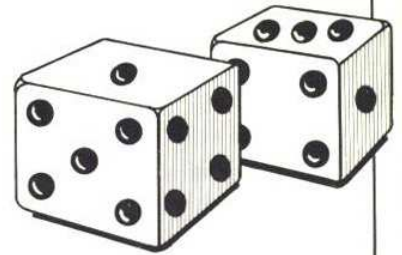
Se o resultado na primeira vez for 7, ou 11, a banca vence, se for 2, 3 ou 12, você vence.

Se não for nenhum desses resultados, o jogo fica indefinido, e jogam-se os dados novamente.

Agora, se a banca repetir o número anterior, você perde. E se a soma dos dados for 7 você vence.

```
   0 REM C-MICROMEGA 1983
1010 LET CT=0
1020 CLS
1030 PRINT
1040 PRINT TAB 9;"JOGO DE CREPE"
1050 PRINT
1060 PRINT "V.SABE AS INSTRUCOES
 (S/N)?"
1070 INPUT A$
1080 IF A$="S" THEN GOTO 1310
1090 CLS
1100 PRINT "INSTRUCOES"
1110 PRINT
1120 PRINT "O CREPE E UM JOGO DE
 DADOS.JOGA -SE COM FICHAS.V.JOG
A CONTRA A  BANCA."
1130 PRINT
1140 PRINT "SE O RESULTADO DOS D
ADOS NA PRI MEIRA JOGADA FOR 7 O
U 11,A BANCA VENCE."
1150 PRINT
1160 PRINT "SE FOR 2,3 OU 12.V.V
ENCE"
1170 PRINT
1180 PRINT "SE NAO FOR NENHUM DE
SSES,O JOGO  FICA INDEFNIDO,E JO
GAM-SE OS    DADOS DE NOVO."
1190 PRINT
1200 PRINT "PRESSIONE A TECLA <N
EW LINE>     P/CONTINUAR."
1210 INPUT B$
1220 CLS
1230 PRINT "INSTRUCOES (CONTINUA
CAO):"
1240 PRINT
1250 PRINT "AGORA,SE A BANCA REP
ETIR O PONTO ATERIOR,V.PERDE."
1260 PRINT
1270 PRINT "E SE A SOMA DOS DADO
S FOR 7,V.   GANHA."
1280 PRINT
1290 PRINT "PRESSIONE A TECLA <N
EW LINE>     P/COMECAR A JOGAR."
1300 INPUT B$
1310 LET F=10
1320 CLS
1330 PRINT "BEM VINDO A MESA DE
CREPE."
1340 PRINT
1350 PRINT "VOCE TEM ";F;" FICHA
S."
```

```
1360 PRINT
1370 PRINT "VOCE TEM ";CT;" CRUZ
EIROS."
1380 PRINT
1390 PRINT "CADA FICHA VALE 100
CRUZEIROS."
1400 PRINT
1410 PRINT "QUER TROCAR FICHA P/
DINHEIRO       (S/N)  ?"
1420 INPUT A$
1430 IF A$="S" THEN GOSUB 1970
1440 PRINT
1450 PRINT "QUER TROCAR DINHEIRO
 P/FICHA       (S/N)  ?"
1460 INPUT A$
1470 IF A$="S" THEN GOSUB 2080
1480 PRINT
1490 PRINT "QUER APOSTAR (S/N) ?
"
1500 INPUT A$
1510 IF A$<>"S" THEN GOTO 1890
1520 PRINT
1530 PRINT "QUANTAS FICHAS"
1540 INPUT A
1550 IF A>F THEN GOTO 1530
1560 IF A<0 THEN GOTO 1530
1570 IF A=0 THEN GOTO 1890
1580 GOSUB 2330
1590 DIM D(2)
1600 LET D(1)=D1+D2
1610 IF D(1)=7 OR D(1)=11 THEN G
OTO 1630
1620 GOTO 1650
1630 GOSUB 2210
1640 GOTO 1850
1650 IF D(1)=2 OR D(1)=3 OR D(1)
=12 THEN GOTO 1670
1660 GOTO 1690
1670 GOSUB 2270
1680 GOTO 1850
1690 PRINT
1700 PRINT "JOGO INDEFINIDO.PRES
SIONE A TE   CLA <NEW LINE> P/JOG
AR NOVAMENTE OS DADOS."
1710 INPUT B$
1720 GOSUB 2330
1730 LET D(2)=D1+D2
1740 IF D(1)=D(2) THEN GOTO 1760
1750 GOTO 1780
1760 GOSUB 2210
1770 GOTO 1850
1780 IF D(2)=7 THEN GOTO 1800
1790 GOTO 1820
1800 GOSUB 2270
1810 GOTO 1850
1820 LET D(1)=D(2)
1830 GOTO 1690
1840 PRINT
1850 PRINT "QUER CONTINUAR JOGAN
DO (S/N)  ?"
1860 INPUT A$
1870 CLS
1880 IF A$="S" THEN GOTO 1320
1890 PRINT "SEU SALDO E DE ";F;"
 FICHAS"
1900 PRINT
1910 PRINT "E DE ";CT;" CRUZEIRO
S."
1920 PRINT
1930 PRINT "TENHA UM BOM DIA"
```

▷

```
1940 PRINT TAB 12;"TK CROUPIER."
1950 STOP
1960 REM SUB FICHA/DINHEIRO
1970 CLS
1980 PRINT "QUANTAS FICHAS ?"
1990 INPUT C
2000 IF C>F OR C<0 THEN GOTO 197
0
2010 LET CR=C*100
2020 LET CT=CT+CR
2030 LET F=F-C
2040 CLS
2050 PRINT "V.TEM ";F;" FICHAS E
 ";CT;" CRUZEIROS."
2060 RETURN
2070 REM SUB DINHEIRO/FICHA
2080 CLS
2090 PRINT "QUANTOS CRUZEIROS ?"
2100 INPUT D
2110 IF D>CT OR D<0 THEN GOTO 20
80
2120 LET FP=INT (D/100)
2130 LET CT=CT-D
2140 LET F=F+FP
2150 CLS
2160 PRINT "V.TEM ";F;" FICHAS E
 ";CT;" CRUZEIROS."
2170 RETURN
2180 REM SUB ERRO
2190 PRINT "VALOR ILEGAL.POR FAV
OR REESCREVA"
2200 RETURN
2210 REM SUB PERDEDOR
2220 PRINT
2230 PRINT "A BANCA VENCE."
2240 PRINT
2250 LET F=F-A
2260 RETURN
2270 REM SUB VENCEDOR
2280 PRINT
2290 PRINT "A BANCA PERDE."
2300 PRINT
2310 LET F=F+A
2320 RETURN
2330 REM SUB DADOS
2340 LET D1=INT (6*RND)+1
2350 LET D2=INT (6*RND)+1
2360 CLS
2370 PRINT "O RESULTADO DOS DADO
S E..."
2380 PRINT AT 5,10;D1
2390 PRINT AT 10,10;D2
2400 RETURN
2410 SAVE "CREP■"
2420 GOTO 1000
2430 STOP
```

PROGRAMA DO LEITOR

Tabela Price

Samuel Ejchel

Como sugestão e complementação ao problema do Ministro, queria lembrar que a conhecida Tabela Price, muito usada em negócios imobiliários, é justamente calculada com base em capitalização instantâneas. Assim o programinha anexo "Tabela Price" (fig. 1) pode ainda ser de utilidade prática para corretores, e merece ser divulgado. Com ele pode-se calcular, inclusive, a taxa para frações de ano, se "entrarmos" com N sendo a fração do ano.

Exemplo: para achar o fator da Tabela Price para a prestação que amortiza uma dívida em 1Ø meses, entraríamos com N = 1Ø/12. Note-se porém, que os resultados são bastante próximos, mas não coincidem exatamente com os tabelados. O valor da "taxa ano" deve ser dado como um número decimal, ou seja, se a taxa for 9% a.a., o valor a alimentar será .Ø9.

```
   5 REM TABELA PRICE
  10 PRINT "TAXA ANUAL=";
  20 INPUT I
  30 PRINT I
  40 PRINT "ANOS=";
  50 INPUT N
  60 PRINT N
  70 PRINT "T.PRICE=";
  80 PRINT EXP (I*N)*(I/12)/(((1
+I/12)**(12*N))-1)

EXEMPLO:
TAXA ANUAL=.09
ANOS=3
T.PRICE=.031831787
```

fig. 1

PROGRAMA DO LEITOR

O TK no Direito Trabalhista e Cível

Dr. João Carlos Normanha Salles

Os programas são aplicativos tanto para a área de Direito Trabalhista quanto Cível, pois uma das maiores dificuldades dos advogados militantes é a realização da liquidação processual, ou seja, transformar o direito reconhecido pela justiça de seus clientes no correspondente em dinheiro a ser recebido da parte condenada.

Quando o usuário fornece a data do ajuizamento, a data presente, o valor da parcela, o programa aplica todos os dispositivos legais para atualização desta parcela. Posteriormente, quando não houver mais parcela alguma a ser atualizada, o programa oferecerá um resumo detalhado de todas as atualizações que foram feitas no mesmo processo e ainda a quantia a ser recolhida a título de imposto de renda retido na fonte para Receita Federal.

fig. Ø e 1

```
MES   = 11          ANO   = 1982
JUROS CALCULADOS EM 2 MESES
PRINCIPAL   = CR$ 100000
INDICE   = 1.7
--------------------------------
VALOR ATUAL =   CR$ 170000
TAXA DE 1 POR CENTO
JUROS = CR$ 1700
VALOR TOTAL CR$ 171700


      [illegible]


            [illegible]
[illegible]
DO PRINCIPAL =CR$ 220000
DO CORRIGIDO = CR$ 368000
DOS JUROS = CR$ 9200
[illegible]=CR$ 377200

I.R.(DARF)= CR$ 459

       [illegible]
```

fig. Ø

```
   1 REM "TRABALHISTA"
   2 REM DR. J C NORMANHA SALLES
   3 CLS
   4 PRINT       "ATUALIZACAO TRAB
ALHISTA"
   5 PRINT
  10 PRINT "TECLE MES ATUAL E <N
EW LINE> "
  20 INPUT A
  30 PRINT
  40 PRINT "TECLE ANO ATUAL E <N
EW LINE>"
  50 INPUT B
  60 PRINT
  70 PRINT
  80 PRINT "CONFIRA: MES= ";A;"
ANO= ";B
  90 PRINT
  93 PRINT
  95 PRINT
 100 PRINT
 110 PRINT "TECLE MES AJUIZAMENT
O E       <NEW LINE>"
 120 INPUT C
 130 PRINT
 140 PRINT "TECLE ANO AJUIZAMENT
O E       <NEW LINE>"
 150 INPUT D
 160 PRINT
 165 PRINT
 170 PRINT "CONFIRA: MES= ";C;"
ANO= ";D
 180 PAUSE 200
 190 CLS
 200 PRINT AT 10,0;"JA ESTOU PRO
NTO PARA ATUALIZAR
                     AS PARCELAS:
TECLE OS DADOS
                     SOLICITADOS
E <NEW LINE>"
 202 LET Q=0
 204 LET R=0
 206 LET S=0
 208 LET T=0
 210 PAUSE 200
 320 CLS
 330 LET E=((B-D)*12)+(A-C)
 340 PRINT "MES ?"
 350 INPUT F
 360 PRINT AT 0,4;" = ";F
 570 PRINT AT 0,15;"ANO ?"
 380 INPUT G
 390 PRINT AT 0,19;" = ";G
 400 LET H=((B-G)*12)+(A-F)
 410 IF H>E THEN LET I=E
 420 IF E>=H THEN LET I=H
 430 IF I=1 THEN PRINT AT 2,0;"J
UROS CALCULADOS EM 1 MES"
 440 IF I>1 THEN PRINT AT 2,0;"J
UROS CALCULADOS EM ";I;" MESES"
 450 PRINT AT 4,0;"PRINCIPAL ?"
 460 INPUT J
 470 PRINT AT 4,10;" = CR$ ";J
 480 PRINT AT 6,0;"INDICE ?"
```

fig. 1

```
 450 INPUT K
 500 PRINT AT 6,7;" = ";K
 510 PRINT AT 8,0;"------------
----------------------"
 520 LET L=INT (100*(J*K))/100
 530 PRINT AT 10,0;"VALOR ATUAL
=  CR$ ";L
 540 LET M=INT ((I/2)*(L))/100
 545 LET O=I/2
 550 PRINT AT 14,0;"JUROS = CR$
";M
 555 PRINT AT 12,0;"TAXA DE ";O;
" POR CENTO"
 560 LET N=L+M
 570 PRINT AT 16,0;"VALOR TOTAL
CR$ ";N
 580 LET Q=Q+J
 590 LET R=R+L
 600 LET S=S+M
 610 LET T=T+N
 620 LET U=INT (S*0.05)
 630 PRINT AT 20,0;"      OUTRA
PARCELA ? S/N "
 640 INPUT Z$
 650 IF Z$="S" THEN GOTO 320
 660 IF Z$="N" THEN GOTO 700
 700 CLS
 710 PRINT AT 0,11;" RESUMO "
 720 PRINT AT 2,0;" SOMAS "
 730 PRINT AT 4,0;"DO PRINCIPAL
=CR$ ";Q
 740 PRINT AT 6,0;"DO CORRIGIDO
= CR$ ";R
 750 PRINT AT 8,0;"DOS JUROS = C
R$ ";S
 760 PRINT AT 10,0;" TOTAL =CR$
";T
 770 PRINT AT 14,0;"I.R.(DARF)=
CR$ ";U
 780 PRINT AT 20,0;"      OUTRO
PROCESSO ? S/N "
 790 INPUT Y$
 800 IF Y$="S" THEN GOTO 3
 810 IF Y$="N" THEN CLS
 820 PRINT AT 10,10;"FIM"
```

fig. 1

O programa seguinte diz respeito a atualização de processos cíveis, que recentemente pela lei 6899 de abril de 1981 passaram também a ser objeto de correção monetária. Esta correção é devida somente a partir da vigência da lei segundo regulamentação feita. Por esta razão o programa está, inclusive, em obediência a este dispositivo legal preparado para rejeitar informações anteriores à vigência da lei (fig. 2).

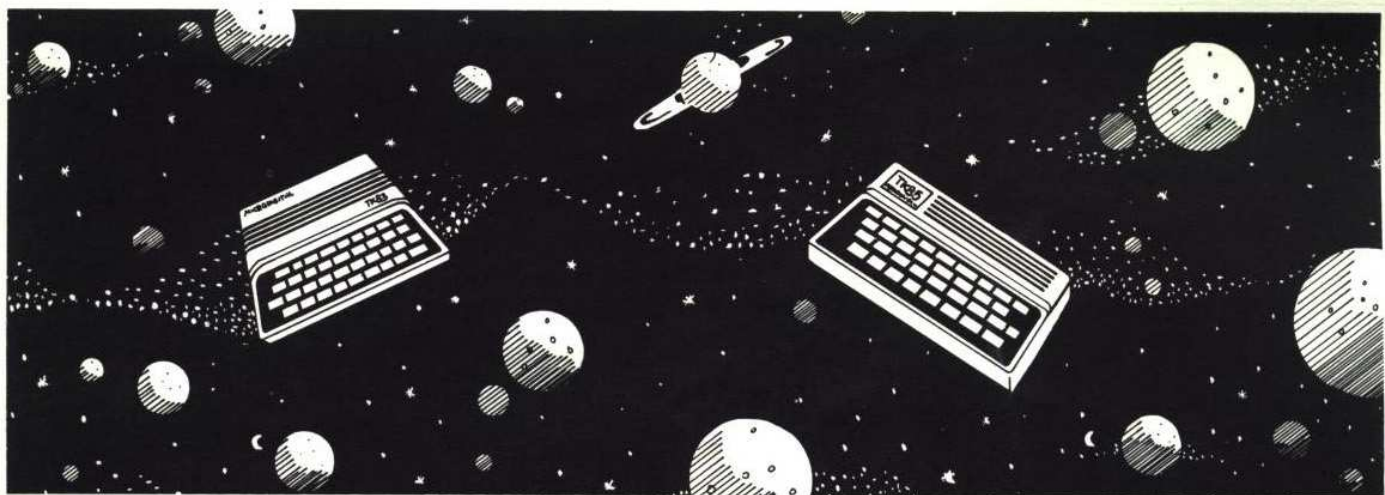
```
MES 12     ANO 1982

JUROS NO PERIODO DE 5 MESES

NA TAXA DE 2.5 POR CENTO

ORTN NA EPOCA = CR$ 2800

ORTN ATUAL = CR$ 4000

PRINCIPAL DE CR$ 100000

********************************

VALOR ATUAL CR$ 142857.14

JUROS VALEM CR$ 3571.42

SOMA CR$ 146428.56

      OUTRA PARCELA ? S/N
```

```
             RESUMO

 SOMAS

DO PRINCIPAL = CR$ 200000

DO CORRIGIDO = CR$ 342857.14

DOS JUROS = CR$ 8571.42

 TOTAL =CR$ 351428.56

OU 87.8571 ORTN

********************************
HONORARIOS DE 50 POR CENTO
NO VALOR DE CR$ 175714.28 OU
43.9285 ORTN
COM I.R. DE CR$ 8785.7
E LIQUIDO DE CR$ 166928.58

     OUTRO PROCESSO ? S/N
```

fig. 2

▷

```
   1 REM "CIVEL"
   2 REM DR. J C NORMANHA SALLES
   5 CLS
   6 PRINT "      ATUALIZACAO CIV
EL"
  10 PRINT
  20 PRINT "TECLE MES ATUAL E  <
NEW LINE>"
  30 INPUT A
  40 PRINT
  50 PRINT "TECLE ANO ATUAL E  <
NEW LINE>"
  60 INPUT B
  70 PRINT
  80 PRINT "CONFIRA: MES ";A;" -
 ANO ";B
  90 PRINT
 100 PRINT "TECLE MES AJUIZAMENT
O E      <NEW LINE>"
 110 INPUT C
 120 PRINT
 130 PRINT "TECLE ANO AJUIZAMENT
O E      <NEW LINE>"
 140 INPUT D
 150 PRINT
 160 PRINT "CONFIRA: MES ";C;" -
 ANO ";D
 170 PRINT
 180 PRINT "VALOR ORTN ATUAL = C
R$";
 185 INPUT M
 186 PRINT M
 187 PRINT
 189 PAUSE 200
 190 PRINT
 195 CLS
 200 PRINT "ESTOU PRONTO PARA CO
RRIGIR AS"
 210 PRINT
 220 PRINT "PARCELAS. TECLE OS D
ADOS QUE"
 230 PRINT
 240 PRINT "FOREM SOLICITADOS E
<NEW LINE>"
 242 LET R=0
 244 LET S=0
 246 LET T=0
 248 LET U=0
 250 PAUSE 300
 260 CLS
 270 PRINT AT 0,0;"MES ?"
 280 INPUT E
 290 PRINT AT 0,4;E
 300 PRINT AT 0,10;"ANO ?"
 310 INPUT F
 320 PRINT AT 0,14;F
 330 LET G=((B-F)*12)+(A-E)
 340 LET H=((B-D)*12)+(A-C)
 350 IF H>=G THEN LET I=G
 355 IF H<=G THEN LET I=H
 360 IF I=1 THEN PRINT AT 2,0;"J
UROS NO PERIODO DE UM MES"
 370 IF I>1 THEN PRINT AT 2,0;"J
UROS NO PERIODO DE ";I;" MESES"
 380 LET J=I/2
 390 PRINT AT 4,0;"NA TAXA DE ";
J;" POR CENTO"
 400 PRINT AT 6,0;"ORTN NA EPOCA
 = CR$ ?"
 410 INPUT K
 420 IF K<877.86 THEN LET L=877.
86
 425 IF K<877.86 THEN PRINT AT 6
20;"877.86(4/81)"
 430 IF K>877.86 THEN LET L=K
 440 PRINT AT 6,20;L;
 450 PRINT AT 8,0;"ORTN ATUAL =
CR$ ";M
 470 PRINT AT 8,17;M
 480 PRINT AT 10,0;"PRINCIPAL DE
 CR$ ?"
 490 INPUT N
 500 PRINT AT 10,17;N
 510 PRINT AT 12,0;"***********
*******************"
 520 LET O=INT (100*(N*M)/L)/100
 530 PRINT AT 14,0;"VALOR ATUAL
CR$ ";O
 540 LET P=INT (O*J)/100
 550 PRINT AT 16,0;"JUROS VALEM
CR$ ";P
 560 LET Q=O+P
 570 PRINT AT 18,0;"SOMA CR$ ";Q
 572 LET R=R+N
 574 LET S=S+O
 576 LET T=T+P
 578 LET U=U+Q
 580 PRINT AT 20,0;"      OUTRA
PARCELA ? S/N"
 590 INPUT Y$
 600 IF Y$="S" THEN GOTO 260
 610 IF Y$="N" THEN GOTO 650
 650 CLS
 660 PRINT AT 0,12;"RESUMO"
 670 PRINT AT 2,0;"SOMAS"
 680 PRINT AT 4,0;"DO PRINCIPAL
= CR$ ";R
 690 PRINT AT 6,0;"DO CORRIGIDO
= CR$ ";S
 700 PRINT AT 8,0;"DOS JUROS = C
R$ ";T
 710 PRINT AT 10,0;"TOTAL =CR$
";U
 720 LET AE=INT ((10000*U)/M)/10
000
 730 PRINT AT 12,0;"OU ";AE;" OR
TN"
 735 PRINT AT 13,0;"***********
*******************"
 740 PRINT AT 14,0;"PERCENTUAL H
ONORARIO ?"
 750 INPUT W
 760 PRINT AT 14,0;"HONORARIOS D
E ";W;" POR CENTO"
 770 LET AB=INT (U*W)/100
 780 LET AC=INT (AB)*0.05
 790 LET AD=INT ((10000*AB)/M)/1
0000
 800 PRINT AT 15,0;"NO VALOR DE
CR$ ";AB;" OU"
 810 PRINT AT 16,0;AD;" ORTN"
 815 PRINT AT 17,0;"COM I.R. DE
CR$ ";AC
 816 LET X=AB-AC
 817 PRINT AT 18,0;"E LIQUIDO DE
CR$ ";X
 820 PRINT AT 20,0;"      OUTRO P
ROCESSO ? S/N"
 830 INPUT Z$
 840 IF Z$="S" THEN GOTO 5
 850 IF Z$="N" THEN CLS
 860 PRINT AT 11,11;"FIM"
```

Ambos os programas contam os juros a partir da data de ajuizamento da ação.

NOVIDADES

TPLAK AMPLIANDO O UNIVERSO DO TK

Flavio Rossini

O computador surge para auxiliar o homem, como uma máquina que não substitui seus músculos mas sim seu **cérebro,** realizando tarefas complicadas e repetitivas, executando cálculos infalivelmente e com velocidades fantásticas, deixando para o homem tarefas mais nobres.

Sem dúvida o que mais contribuiu para a popularização dos computadores foi o aparecimento dos computadores pessoais, os quais surgiram às centenas . . . e no meio deles, apareceu um pequeno, discreto, simpático e, principalmente, acessível a todos: o TK. . . Assim, ele começou a fazer parte de muitos lares pelo mundo afora, quase sempre com o seguinte pensamento: TK, o meu primeiro computador. . . Mas ele é orgulhoso. Ele diz: "se os outros computadores podem tocar música, eu também posso. Se os outros podem fazer desenhos complicados, eu também posso. Posso até comandar o funcionamento de outras máquinas ou me comunicar por telefone com um dos meus 2.000.000 de irmãos. Posso fazer tudo que qualquer outro faz.

De fato, apesar de não ser tão rápido e tão imponente ele tem o mesmo cérebro (CPU) que os "grandes", além de uma grande vantagem: é acessível, em termos de preço, o que o tornou o computador mais vendido da História. . . Não só, mas ele é versátil podendo **crescer** junto com sua imaginação para fazer mil coisas. O truque está nas conexões traseiras: atualmente, você já pode ligar a impressora e a expansão da memória. Mas e se você pudesse ligar um circuito para fazer sons, ou desenhos, ou até mesmo fazê-lo falar? Ou comandar aparelhos externos, um mini-robô, talvez?. . . Ou simular que tem gente em casa quando você vai viajar, para enganar os ladrões? Tudo isto já é possível com o surgimento dos periféricos TplaK. O que vem a ser eles? São placas de circuito impresso que você conecta diretamente ao seu TK e que, aliadas a sua imaginação, permitirão ao seu microcomputador a realização de façanhas, as quais ele era INJUSTAMENTE julgado incapaz de fazer.

Vamos este mês comentar três placas bastante interessantes:

a) placa de "beep" (BPO-TX3), que produz um sinal sonoro ao toque da tecla oupor comando de programa.
b) placa de entrada/saída (I/O-TX2), que permite o comando de dispositivos externos, enviando e recebendo sinais binários.
c) placa de caracteres (CHR-TX4), que permite criar novos caracteres para o TK.

A primeira delas produz um sinal sonoro cada vez que alguma tecla é pressionada, permitindo maior facilidade para a inserção de programas no computador; além disso, ao fazer em **SAVE** ou **LOAD**, o "beep" toca durante a passagem dos dados do gravador para o computador e ▷

vice-versa, parando no final, permitindo a você executar outras coisas enquanto aguarda a passagem do programa.

você dirige um avião que deve atingir um alvo, produzindo um sinal sonoro ao atingí-lo.

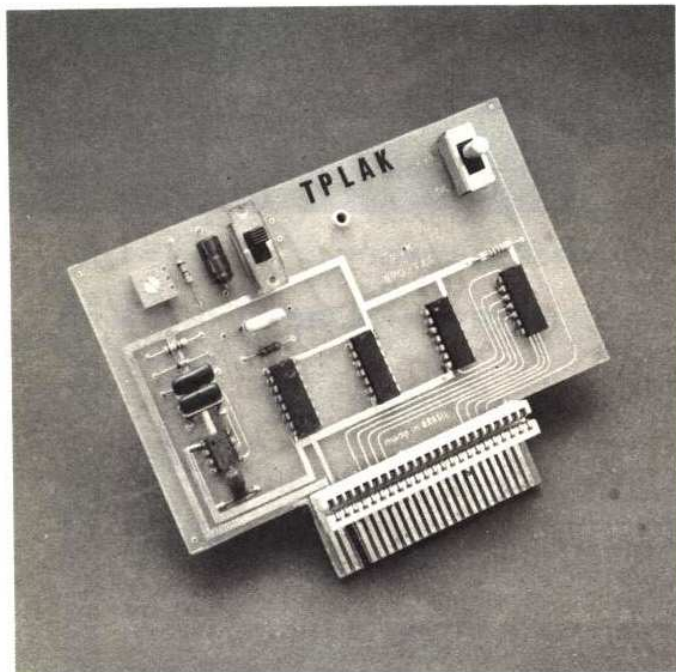



FIGURA 1

PLACA BPO TX3 – SONORA BIP

Mediante sonorização bip, assegura automaticamente ao usuário a entrada do dado correspondente à tecla digitada. Também sonoriza bip por programação, ou seja, "bipa" nos momentos desejados, durante a execução do programa. Auxilia o uso dos comandos **SAVE** e **LOAD**, interrompendo o sinal sonoro ao final das operações.

FIGURA 2

PLACA I/O TX2 – ENTRADA/SAÍDA

Apresenta oito entradas e oito saídas controladas pelo usuário por programação. Possibilita o comando de aparelhos elétricos e/ou eletrônicos, tais como eletro-domésticos, motores elétricos, fontes de alimentação etc. Permite, inclusive, o comando de máquinas industriais como, por exemplo, um torno-revólver. Admite a montagem de várias similares em série, aumentando proporcionalmente o número de entradas e de saídas de controle.

Finalmente, este "beep", que pode ser ligado e desligado à vontade, pode também ser acionado pelo próprio programa. . . Imagine um joguinho onde você faz com que o "beep" toque sempre que um avião atinge um alvo. E já que falamos em jogos, podemos falar sobre a segunda placa: ela permite comandar (ligar ou desligar) até 8 dispositivos externos quaisquer através do seu programa. O ligamento é sinalizado através de 8 leds.

Além disso, ela possui 8 entradas para receber sinais externos. Esqueçamos, por enquanto, o comando de dispositivos externos e vamos pensar apenas nos LEDS. Que tal se no "joguinho do avião" os leds simulassem um painel de controle: falta de combustível, vento muito forte, tiro efetuado . . . Cinco das 8 entradas poderiam ser utilizadas para um joystick que comanda a direção do avião, duas outras entradas para o acelerador. . . Assim, associando essas placas com a de "beep",

Naturalmente, usando a imaginação pode-se chegar aos mais altos níveis de sofisticação. Com a ajuda de relés, pode-se até automatizar uma máquina que tenha motores, indicadores de fim de curso, enfim, quase um mini-robô

Com esta placa, o TK pode ligar motores, disparar sinais de alarme, e ser programado para execução de várias tarefas.

E que tal se esta placa for usada para ligar e desligar "aleatoriamente", 8 luzes da sua casa enquanto você estiver viajando para enganar os ladrões? Ou comandar o seu gravador para facilitar o **SAVE** e o **LOAD**? Ah, e se 8 entradas e 8 saídas forem poucas, várias placas (até 256) podem ser ligadas ao TK, cada uma podendo comandar 8 saídas e receber 8 entradas.

FIGURA 3
PLACA I/O TX2 – ENTRADA/SAÍDA
Placa TX2, montada com placa MB2 TX1 (Placa Mãe), em computador TK85.

Mas, voltando ao joguinho, se você quisesse um avião mesmo na tela e não apenas um desenho utilizando os caracteres normais do TK? Ou se você quisesse jogar xadrez, utilizando o desenho real das peças na tela e não apenas suas iniciais? Ou escrever utilizando letras minúsculas, ou letras gregas, fórmulas químicas . . . , quem sabe fazer um jogo de cartas e poder desenhar os quatro naipes? Tudo isso é possível com a placa de caracteres que permite a criação de qualquer caracteres à sua escolha. Esta placa possui uma memória RAM na qual você pode colocar os modelos dos caracteres que deseja.

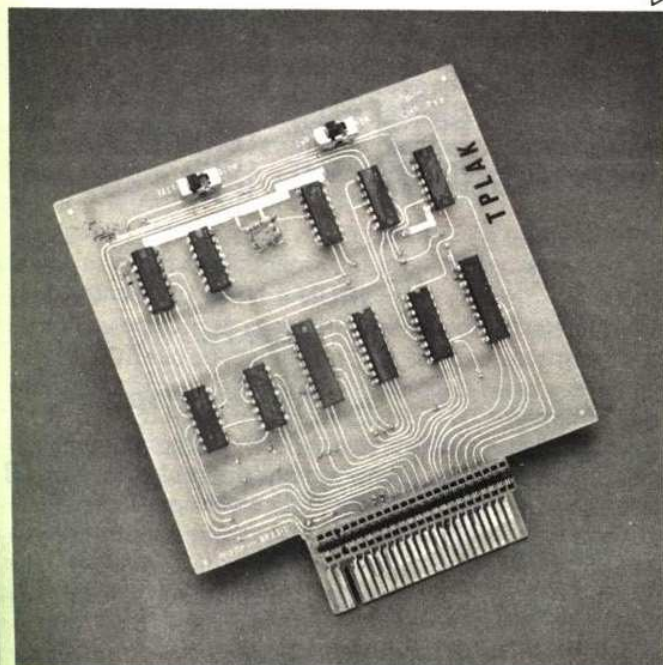



FIGURA 4
PLACA CHR TX4 – CARACTERES
Transforma os caracteres da lógica Sinclair em caracteres criados pelo usuário. É acompanhada por programa em fita cassete, possibilitando seu uso imediato. Equipada com memória de 1 Kb para a armazenagem dos caracteres criados pelo usuário, possibilitando sua utilização em diferentes programas.

Naturalmente, "compondo caracteres", você poderá até fazer desenhos de alta resolução. . .

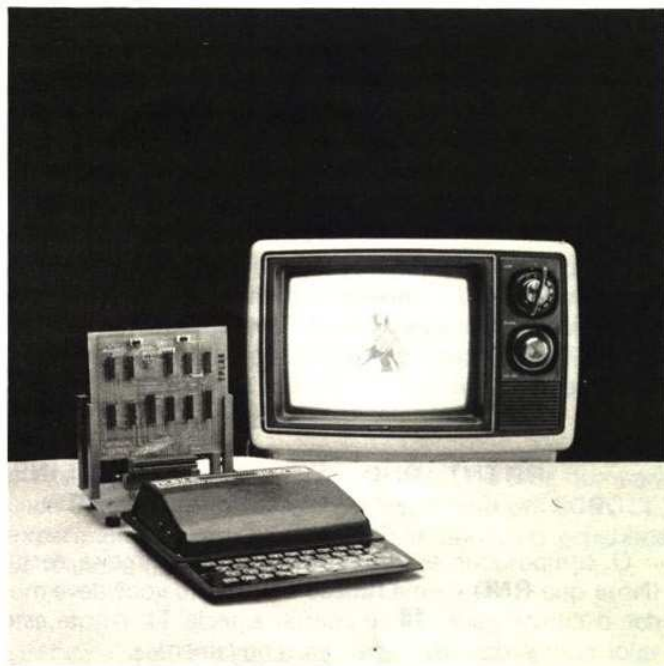



FIGURA 5
PLACA CHR TX4 – CARACTERES
A foto mostra uma aplicação da Placa CHR TX4. Neste caso, o usuário usou a criação de novos caracteres para desenhar na tela essa cabeça de cavalo. A montagem da CHR TX4 é sobre uma Placa MB2 TX1 (Placa Mãe) conectada a um computador TK82-C.

Realmente estas placas ampliam o já gigantesco universo do TK, tirando qualquer "complexo de inferioridade" que ele pudesse ter em relação a seus irmãos maiores.

CURSO DE BASIC TK

AULA 3

Pierluigi Piazzi
Flavio Rossini

Vamos aprender agora uma função muito interessante que possibilita o computador "chutar" números; para entendê-la, imagine que dentro do computador existam INFINITAS cartas, cada carta possuindo um número compreendido entre Ø e 1; se você ordenar:

```
PRINT RND
```
(**NEW LINE**)

O computador escreverá o valor da primeira carta. (Note que **RND** é uma **função** e portanto você deve mudar o cursor para F e apertar a tecla T). Anote este valor num pedaço de papel. Faça novamente:

```
PRINT RND
```

Anote o novo valor (que corresponde à segunda carta) e repita a operação; você terá então o valor das primeiras três cartas. Desligue o computador; ligue-o novamente e repita a operação; você obterá novamente as primeiras 3 cartas. No entanto, existe um comando que faz com que o computador "CHUTE" qual a primeira carta que ele deve escolher deste monte INFINITO, estando as próximas em seguida; dessa forma, você obterá sempre sequências diferentes. Este comando é a KEY-WORD **RAND**; faça:

```
RAND
```
(**NEW LINE**) (na tecla T)

e a seguir execute a operação acima. Desligue e ligue o computador faça **RAND** e repita. Desta vez os números serão diferentes (Se, por coincidência do destino, isto não ocorrer, repita a operação. Dificilmente você irá conseguir isto 3 vêzes seguidas. . .).

Ordene agora o seguinte:

```
PRINT 100*RND
```

ora, o que você espera obter? Novamente um número "chutado" só que desta vez entre Ø e 1ØØ, não mais entre Ø e 1.

Se tirarmos a parte INTEIRA deste número, poderemos usá-lo para jogar na LOTO. Para isto, existe a função **INT** (tecla R):

```
PRINT INT(100*RND)
```

Assim, obtém-se um número entre Ø e 99.

Faça isto 5 vezes e você terá 5 números para jogar na LOTO essa semana. Vamos então agora "programar" o computador para que ele "execute" essas ordens? Coloque então o seguinte programa:

```
10 RAND
20 PRINT "LOTO"
30 PRINT
40 PRINT INT (RND*100)
50 PRINT INT (RND*100)
60 PRINT INT (RND*100)
70 PRINT INT (RND*100)
80 PRINT INT (RND*100)
```

execute-o diversas vêzes (**RUN** e **NEW LINE**). Cada vez em que ele for executado, ele pega as próximas 5 cartas com números entre Ø e 1, multiplica-os por 1ØØ e "despreza" as casas depois da vírgula; assim, cada vez, você terá 5 números INTEIROS entre Ø e 99.

Observe que tivemos que **repetir** 5 vêzes a mesma instrução; imagine se precisássemos repeti-la 1ØØ vêzes. Não seria muito agradável ter que fazer isto. Felizmente, existem duas KEY-WORDS no TK que permitem ordenar ao computador que ele execute diversas vêzes uma (ou mais) instruções. Para isto, imagine que alguém peça para você repetir 5 vêzes uma dada palavra; para fazer isto, cada vez que você diz a palavra, você deve estar "contando" em algum lugar da sua cabeça ("memória") o número de vêzes que você a repetiu. Da mesma forma, se desejarmos que o computador repita uma dada instrução, precisamos dizer em que VARIÁVEL da memória ele pode CONTAR o número de vêzes em que ele a repetiu. A estrutura deve ser a seguinte: ▷

```
FOR variável = nº inicial TO nº final
  INSTRUÇÕES A SEREM REPETIDAS
NEXT variável
```

As Key Words **FOR** e **NEXT** estão nas teclas F e N respectivamente; o **TO** está na tecla 4 (não adianta escrever as letras T e O; deve ser **SHIFT** 4); assim, o programa anterior poderia ser escrito da seguinte maneira:

```
10 RAND
20 PRINT "LOTO"
30 PRINT
40 FOR I=1 TO 5
50 PRINT INT (100*RND)
60 NEXT I
```

Note que usamos a variável **I** para fazer com que ele contasse de 1 até 5. Lembre-se sempre então: para repetir uma ou mais instruções elas devem estar ENTRE um **FOR** e um **NEXT**. A este tipo de estrutura chamamos **LOOP** de repetição. Experimente então executar o programa. Que modificação você deverá fazer se quisesse obter 10 números e não 5?

Neste ponto, várias observações são necessárias:

a) ao terminar um loop de **FOR-NEXT**, a variável contadora estará com um valor de uma unidade maior do que o valor final da instrução de **FOR**. De fato, experimente fazer agora o seguinte comando direto:

```
PRINT I                    (NEW LINE)
```

b) a variável contadora, ao contrário das variáveis usadas com a instrução **LET** pode ter **somente uma letra.** Com a instrução **LET** a variável pode ter quantas letras quisermos, inclusive números, desde que o primeiro caractere seja uma letra.

c) a função **INT** fornece como resultado o número inteiro de valor imediatamente inferior ao seu argumento; assim temos:

```
INT 5.76=5

INT -5.76=-6
```

d) numa expressão matemática, todas as funções (cursor **F**) tem prioridade máxima; assim, por exemplo, **INT** 3.4*X é diferente de **INT** (3.4*X). No primeiro caso, sendo a função prioritária com relação às demais operações, teremos:

```
INT 3.4*X=3*X
```

Agora, retire as linhas 1Ø, 2Ø e 3Ø do programa e substitua as linhas 4Ø e 5Ø de tal maneira que seu programa fique:

```
40 FOR I=1 TO 50
50 PRINT I
60 NEXT I
```

Execute o programa. Veja, você ordenou que ele escrevesse a própria variável onde ele está CONTANDO. No entanto, como ele só tem 22 linhas disponíveis, irão aparecer na tela os números de 1 a 22. Mas nós queríamos que ele contasse até 5Ø; ora, basta mandá-lo continuar. Pressione:

```
CONT                    (tecla C) e NEW LINE
```

e veja o que acontece. Repita novamente (quando o programa parar) até obter os 50 números.

Vejamos agora como executar o programa anterior sem que ele pare. Note que os números são "impressos" de cima para baixo, parando ao chegar no fim da tela; existe um comando que faz com que os números sejam escritos de baixo para cima **sem** interromper o programa. Este comando é a **KEY-WORD SCROLL** (tecla B) e ao ser colocado ANTES do **PRINT**, ele simplesmente "rola" a tela para "cima"; assim acrescente a linha 45:

```
45 SCROLL
```

e execute o programa. Veja o efeito. Perceba que para cada **PRINT** que colocarmos é necessário um **SCROLL**; experimente acrescentar mais duas linhas ao programa de maneira que ele fique:

```
40 FOR I=1 TO 50
45 SCROLL
50 PRINT I
53 SCROLL
56 PRINT "PAZ"
60 NEXT I
```

Se você não colocar o segundo **SCROLL**, o programa não funcionará. Execute então o programa. Se você desejar "pará-lo" antes que ele termine (não vale desligar o computador) basta apertar a tecla **BREAK** (a mesma do **SPACE**). Para continuar, é só apertar **CONT**; note que este último sempre LIMPA a tela antes de mandar continuar o programa.

Neste ponto, podemos falar um pouco sobre os números que aparecem no canto inferior esquerdo da tela quando o programa para; o número à esquerda indica o "por quê" do programa ter parado; o número à direita, indica a linha onde o programa parou. Assim, o código Ø indica tudo OK; se você deixar o programa acima terminar "sozinho" você obterá o código:

```
0/60
```

(tudo OK; programa parou na linha 6Ø).

O código D indica que o programa parou devido a um **BREAK**; se você parar o programa assim, irá obter:

D / número que depende do instante em que você apertou a tela

▷

Estes códigos aparecem no fim do manual do TK; no decorrer do curso, procuraremos analisá-los todos.

Vamos agora relembrar o efeito do ";". Retire as linhas 45 e 53 e, a seguir, modifique a linha 5Ø para:

```
50 PRINT I;
```

e execute o programa.

A seguir, acrescente um; também no final da linha 56:

```
56 PRINT "PAZ";
```

Lembre-se bem do efeito do ";" que é informar ao computador que o próximo **PRINT** deverá ser feito ao lado da última coisa que foi impressa.

Vamos melhorar um pouco a tela, modifique as linhas 5Ø e 56 de tal maneira que o programa fique:

```
40 FOR I=1 TO 50
50 PRINT I;" ";
56 PRINT "PAZ ";
60 NEXT I
```

Lembre-se que para obter o espaço em branco deve-se apertar a tecla **SPACE**. Após executar o programa apague-o (usando **NEW**) e vamos introduzir uma nova função: **TAB** (letra P) que, analogicamente a uma máquina de escrever, faz uma tabulação, ou seja, faz com que a impressão comece na coluna que desejarmos; experimente fazer:

```
10 FOR I=1 TO 50
20 PRINT TAB 10;I
30 NEXT I
```

Note que os números são escritos a partir da coluna 1Ø. O que aconteceria se modificassemos a linha 2Ø para:

```
20 PRINT TAB I;1
```

Faça esta modificação e veja o que ocorre. Você está usando uma variável para fazer a tabulação. Acrescente então a linha 15:

```
15 SCROLL
```

e note o efeito. Note que quando o argumento do **TAB** for maior que 31, o computador "dá a volta" para recomeçar a escrever no início da tela (para perceber isto no 1º programa você deverá fazer dois **CONTs**). Para firmar bem o efeito do **TAB** e do **SCROLL**, altere novamente a linha 2Ø:

```
20 PRINT TAB I;I;TAB (2*I);"X"
```

execute o programa deixando a linha 15 **SCROLL** e a seguir retirando-a. (Neste caso, não esqueça do CONT).

Perceba que se o argumento da função **TAB** não for inteiro, o computador automaticamente calcula a parte inteira para fazer a impressão na coluna certa. O mesmo é válido para a função **AT**.

Vejamos agora o efeito que a "," (vírgula) causa na impressão; modifique a linha 2Ø de tal maneira a termos o programa:

```
10 FOR I=1 TO 50
20 PRINT I,
30 NEXT I
```

Note que a vírgula desloca a próxima posição de **PRINT** de 1/2 tela. O que ocorre se você modificar a linha 2Ø para:

```
20 PRINT I,,
```

Tente repetir o processo para 3, 4 e 5 vírgulas e repare no efeito.

Para finalizar as "palavras novas" deste capítulo, vamos supor que quisessemos uma "tela" da seguinte maneira:

Isto pode ser feito de maneira análoga à que fizemos agora há pouco para escrever na outra diagonal, se o computador contar de 21 para Ø, ou seja, em contagem regressiva; para isto faça:

```
10 FOR I=21 TO 0 STEP -1
20 PRINT TAB I;I
30 NEXT I
```

A palavra **STEP** está na tecla E, e ela significa de "quanto em quanto" o computador deve contar. Note que ao ser omitida como fizemos até agora o computador assume **STEP** igual a 1, ou seja, conta de 1 em 1; no caso, vai de −1 em −1; o **STEP** pode ter qualquer valor; experimente mudá-lo para −2.

Finalizando, vamos apresentar um pequeno programa para recordar o que vimos até aqui inclusive a função **AT**:

▷

```
1Ø RAND
2Ø LET N = 1Ø
3Ø FOR I = 1 TO N
4Ø LET A = INT (RND * 22)
5Ø LET B = INT (RND * 32)
6Ø PRINT AT A, B; I
7Ø NEXT I
```

Repare que a linha 4Ø "gera" um número aleatório entre Ø e 21 e que a linha 5Ø um número entre Ø e 31 que correspondem respectivamente às linhas e colunas da tela; assim iremos escrever os números de 1 a 1Ø em lugares "aleatórios" da tela. Experimente a seguir substituir a linha 2Ø por:

```
2Ø LET N = 2Ø
```

E execute novamente o programa.

Exercícios:

1) Faça um programa que produza o efeito diagonal começando à direita em cima (como no penúltimo programa) SEM utilizar **STEP**.

2) Faça um programa que escreva várias vêzes a palavra AMOR em vídeo reverso (que tenha no começo e no fim um caractere de **SPACE** também em vídeo reverso) em diagonal, começando no canto superior esquerdo.

3) Faça um programa que gere 13 números aleatórios entre 1 e 3 e apresente a seguinte saída:

```
LOTERIA

1
2
2
3
2     13 nºs aleatórios
1
2
3
.
.
.
```

a seguir, implemente a saída para que ela fique assim:

LOTERIA	
Jogo 1	Coluna 2
Jogo 2	Coluna 3
.	.
.	.
.	.
Jogo 9	Coluna 1
Jogo 1Ø	Coluna 2
Jogo 11	Coluna 3
Jogo 12	Coluna 1
Jogo 13	Coluna 2

Note que a palavra COLUNA **não** deve aparecer deslocada nos jogos 1Ø, 11, 12 e 13. Finalmente, estude um método para que a saída na tela seja:

LOTERIA	Col. 1	Empate	Col. 2
Jogo 1	X		
Jogo 2		X	
Jogo 3		X	
.			
.			
.			
Jogo 13			X

4) Faça um programa que "encha" a tela de asteriscos (*) a partir da posição 1Ø, 1Ø.

CURSO DE ASSEMBLY

AULA 2

Flavio Rossini

Este curso foi escrito para as pessoas que já estejam bastante familiarizados com o BASIC-TK mas que tenham pouco ou nenhum conhecimento sobre linguagem de máquina.

Iremos inicialmente, introduzir o conceito de "microprocessador". O computador como nós conhecemos consiste dos seguintes elementos principais:

- o microprocessador (modelo Z8Ø)
- a memória
- dispositivos de entrada (teclado, joystick)
- dispositivos de saída (vídeo, impressora)
- memória auxiliar (gravador K7)

O computador pode ser comparado por analogia a um ser humano, só que ele "fala" BASIC em vez de português. Assim, esse "ser" tem um **cérebro** (que é o **microprocessador**) e uma memória. Você conversa com ele através do teclado (para enviar informações) e da tela (para receber informações), usando o BASIC. Acontece que o **microprocessador NÃO entende BASIC . . .** Ele entende uma língua muito mais simples e limitada à qual chamamos de **linguagem de máquina.** Dessa forma, parte da memória do computador contém um programa escrito em **linguagem de máquina** capaz de TRADUZIR o BASIC, para que o microprocessador entenda os comandos, e, vice-versa, capaz de traduzir as respostas fornecidas pelo microprocessador.

Nosso intuito será o de conseguir conversar DIRETAMENTE com o **microprocessador** na sua **própria linguagem.** É desejável para isto que você esteja bem familiarizado com o funcionamento das instruções **PEEK** e **POKE** e com os números escritos em binário ou hexadecimal (caso você não conheça estes conceitos, comece lendo a aula 1 (MICROHOBBY nº 2).

Além disso, vamos começar a entender como funciona o programa tradutor, utilizando para nossos programas algumas subrotinas já existentes ou valores de algumas variáveis por ele utilizadas para controle do sistema.

Vários programas serão apresentados e já na primeira parte do curso conseguiremos realizar um **SCROLL** ao contrário, passando a truques para lidar com a TV e o teclado e finalizando com um incrível programa chamado **LIFE**, cuja listagem estamos publicando neste número da revista.

Estude com cuidado CADA programa e procure responder a TODAS as questões que acompanham o texto antes de prosseguir a leitura. No final de cada aula haverá eventualmente exercícios para serem resolvidos que foram feitos não só para recordar os conceitos aprendidos mas também com o intuito de introduzir alguns "truques" e conceitos muito importantes. Portanto, sugiro que você faça os exercícios de cada aula antes de prosseguir para o seguinte.

Seria aconselhável que você tivesse uma expansão de memória de 16 K para que possa executar os programas mais interessantes deste livro, embora ela não seja estritamente necessária para aprender a linguagem de máquina.

Você poderia se perguntar: o que esta linguagem tem para me oferecer? A resposta resume-se a duas palavras: RAPIDEZ e CONTROLE, e você perceberá logo mais.

Bom, seja bem vindo ao Universo da Linguagem de Máquina.

O QUE É A LINGUAGEM DE MÁQUINA; O PROGRAMA INTERPRETADOR

A linguagem de máquina (também chamada de ASSEMBLY) é a "língua" que o microprocessador entende. Ela consiste, assim como o BASIC, de uma série de instruções, cada uma das quais comanda o microprocessador para que ele faça uma tarefa específica. Entretanto, ela é uma linguagem de baixo nível, o que significa que instruções complicadas como loops de FOR/NEXT ou cálculos de funções matemáticas (SIN, COS, EXP, RND, etc.) não são disponíveis; pode-se usar apenas instruções que comandam **diretamente** o microprocessador.

O que significa exatamente "linguagem de baixo nível" com instruções que comandam "diretamente" o microprocessador?

O problema é que o microprocessador, por ser um circuito eletrônico, só é capaz de entender **dois** níveis de tensão elétrica que chamamos simbolicamente de Ø (zero) e 1 (um). Portanto, somos obrigados a conversar com ele usando **códigos** formados por "cadeias" (sequências) de Øs a 1s, onde cada Ø ou 1 é chamado BIT. Assim, por exemplo, o código 1ØØØØ1Ø1 poderia significar "some dois números", o código 1Ø1ØØØ1 "pare o programa", etc. Além disso, o microprocessador só é capaz de entender 8 bits (1 byte) por vez. Dessa forma, cada instrução e cada dado deve, obrigatoriamente, consistir de um ou mais grupos de 8 bits. No caso de números, o número máximo que podemos representar em 1 byte é 255. Se o número for maior que 255 ele deverá ser "quebrado" em 2 ou mais bytes para que o microprocessador possa entendê-lo.

Acontece que para o ser humano não é muito fácil conversar em Øs e 1s (basta tentar imaginar uma página cheia de "códigos" em Øs e 1s). Assim convencionou-se que os bits seriam agrupados de 4 em 4 substituindo-os pelos correspondentes algarismos em HEXADECIMAL. ▷

Portanto, se quisermos falar **diretamente** com o **microprocessador,** devemos lembrar que ele só entende bits, e cada instrução consiste em um ou mais bytes que, para maior facilidade de entendimento, são escritos em HEXADECIMAL, correspondendo, então, cada byte a **dois digitos hexadecimais** (por exemplo C9, D4, 37, etc.). (OBS.: para quem não está familiarizado com as notações acima indicadas, sugerimos novamente que, antes de prosseguir, leia a aula 1, números binários e hexadecimais, bit e byte, **PEEK** e **POKE**).

A MEMÓRIA DO COMPUTADOR: ROM E RAM

Assim como as instruções são sempre divididas em grupos de 8 bits, a memória do computador também é dividida em regiões capazes de armazenar apenas 8 bits cada uma.

A memória pode ser imaginada como uma pilha de "registros" capazes de armazenar apenas um byte cada, sendo que cada registro tem um **endereço** que, no caso do TK, pode ir de Ø até 65535 (correspondendo a ØØØØ até FFFF em hexadecimal) (Fig. I.1).

ENDEREÇO DECIMAL	ENDEREÇO HEXADECIMAL	CONTEÚDO DA MEMÓRIA								CONTEÚDO EM HEXADECIMAL
Ø	ØØØØ	Ø	1	1	Ø	1	1	1	1	6F
1	ØØØ1	1	Ø	Ø	1	Ø	1	1	1	97
2	ØØØ2	1	Ø	1	Ø	1	Ø	1	1	AB
3	ØØØ3	Ø	Ø	1	1	Ø	1	Ø	1	35
4	ØØØ4	1	1	1	Ø	1	1	Ø	1	ED
5	ØØØ5	Ø	1	Ø	Ø	1	1	1	Ø	4E
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
65530	FFFA	1	1	Ø	Ø	Ø	Ø	Ø	1	C1
65531	FFFB	Ø	1	Ø	Ø	Ø	Ø	Ø	Ø	4Ø
65532	FFFC	Ø	1	Ø	Ø	Ø	Ø	1	Ø	42
65533	FFFD	1	1	1	1	Ø	1	Ø	1	F5
65534	FFFE	Ø	Ø	1	1	1	Ø	1	Ø	3A
65535	FFFF	1	Ø	1	1	1	Ø	Ø	1	B9

Fig. I.1: Exemplo de como pode ser imaginada uma memória com seus respectivos endereços e conteúdo; note que o conteúdo de um registro de memória tem sempre 1 byte (8 bits) podendo ser representado com 2 dígitos hexadecimais e seu respectivo endereço 2 bytes (16 bits), podendo ser representado com 4 dígitos hexadecimais.

A memória é basicamente dividida em duas partes: a primeira chamada ROM (Ready Only Memory), foi previamente gravada por processos especiais e contém o programa que **interpreta a linguagem BASIC**, ou seja, "traduz as instruções do BASIC para a linguagem de máquina. Esta região da memória é inalterável, não podendo, portanto, ser escrita (apenas lidas) e não perde nunca seu conteúdo, mesmo desligando-se o computador. A segunda, chamada RAM (Random Access Memory) é alterável (pode ser lida ou escriga), e é onde são armazenados o conteúdo da tela de TV, os programas em BASIC devidamente "traduzidos" pelo interpretador (ROM), as variáveis dos programas e, inclusive, os futuros programas em linguagem de máquina que faremos neste curso. Ela perde seu conteúdo desligando-se o computador.

Note que o programa interpretador (ou tradutor) que está na memória ROM é escrito em **linguagem de máquina,** pois o microprocessador deve ser capaz de entendê-lo. Entretanto, como todo programa, ele possui **variáveis,** que devem ser, obrigatoriamente, colocadas na RAM (caso contrário não seriam variáveis). Assim, ele **reserva** para si o INÍCIO da memória RAM, mais precisamente os endereços **16384 a 165Ø8** para colocar suas próprias variáveis.

Portanto, os programas que fizermos em linguagem de máquina deverão ser colocados em alguma região da memória RAM, inteligentemente escolhida para que não afete o andamento dos demais programas (eventuais programas em BASIC), a própria tela de TV ou as variáveis do programa interpretador. Quando programamos em BASIC estes cuidados não são necessários, pois o programa residente na ROM se encarrega de colocar as instruções em lugares adequados da RAM.

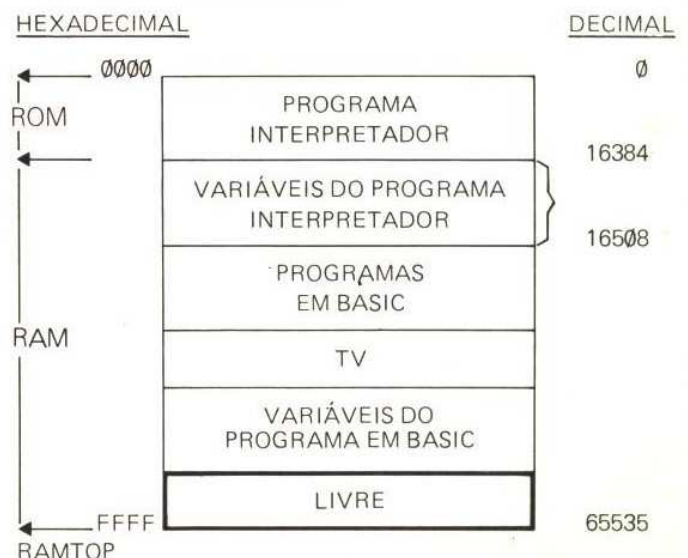



Fig. I.2: A memória do TK – a memória só irá até 65535 se usarmos a expansão de 64 K.

Na próxima aula, veremos então COMO e ONDE podemos colocar um programa em linguagem de máquina no computador.

QUEBRA-CABEÇA

Motoqueiro no Deserto

Era uma vez um motoqueiro, não muito inteligente, que vivia numa aldeia à beira do deserto. Sua moto tinha um tanque no qual cabia combustível suficiente para andar 300 km e na aldeia ele tinha um total de combustível (incluindo o da moto) suficiente para 900 km. O motoqueiro não podia transportar nenhum vasilhame contendo combustível, mas podia levar consigo sacos de plástico vazios para enchê-los com parte do combustível do tanque. Estes sacos, enterrados na areia, formavam um reservatório que poderia ser usado posteriormente.

O motoqueiro, apesar de não muito inteligente, obedecia fielmente às instruções que recebia, por exemplo assim:

```
10 ENCHA O TANQUE NA ALDEIA
20 PENETRE 100 KM NO DESERTO
30 FAÇA UM RESERVATÓRIO COM COMBUSTÍVEL PARA 100 KM
40 VOLTE PARA A ALDEIA
50 ENCHA O TANQUE NOVAMENTE
60 VÁ ATÉ O RESERVATÓRIO
70 COLOQUE O COMBUSTÍVEL DO RESERVATÓRIO NO TANQUE
80 PENETRE MAIS 100 KM
90 PLANTE UMA BANDEIRA
100 VOLTE PARA A ALDEIA
```

Ele cumpriria fielmente as instruções e acabaria plantando uma bandeira a "200 km" da aldeia gastando combustível para "600 km".

Agora, para variar um pouco, ao invés de programar seu TK você vai programar o motoqueiro.

Lembre-se, ele só tem um total de combustível suficiente para 900 km e no tanque da moto cabe o equivalente a 300 km.

Tente fazer o motoqueiro plantar a bandeira o mais longe possível da aldeia (regressando a salvo, obviamente).

Quem mandar o melhor programa para motoqueiro fazendo-o chegar o mais longe possível, receberá o prêmio costumeiro e mais o título de "Raposa do Deserto 83".

RESPOSTA DO QUEBRA-CABEÇA

O Banquiero e o Ministro

Jaime Maia Neto

Para resolver o quebra-cabeças, "O Banqueiro e o Ministro", comecei por fazer o cálculo do número N, proposto no problema, para algumas partições P do ano.

$P = 2 \quad N = 1{,}5 \times 1{,}5 = (1{,}5)^2 = 2{,}25$

$P = 4 \quad N = 1{,}25 \times 1{,}25 \times 1{,}25 \times 1{,}25 =$
$= (1{,}25)^4 = 2{,}44$

$P = 8 \quad N = (1{,}125)^8 = 2{,}57$

$\vdots \qquad \vdots$

A essa altura, já pude ver que a fórmula genérica de N é:

$$N = \left(1 + \frac{1}{P}\right)^P$$

onde P é o número de partes em que está dividido o ano.

Acontece que a fórmula obtida, é muito conhecida do Cálculo. E sabe-se que quando P tende ao infinito, o valor de N tende ao número "**e**" que vale aproximadamente 2,7182818. Escreve-se:

$$\lim_{P \to \infty} \left(1 + \frac{1}{P}\right)^P = e \approx 2{,}7182818$$

Ora, se para um número P tão grande quanto desejarmos, N não ultrapassa 2,72, então os juros não podem ser maiores que 172%, por mais pequena que seja a unidade de tempo usada para a cobrança de juros.

Em seguida fiz um programa (fig. 1) que mostra que quando aumentamos o valor de P, N tende a "e".

P programa começa com um P = 1; depois P = 2; P = 4; P = 8; P = 16 . . . ; mostra a unidade de tempo T que se está tomando para a cobrança dos juros e o valor de N.

```
   1 REM O BANQ. E O MINISTRO
  10 LET PART=1000*60*60*24*365
  20 LET P=1
  30 LET N=INT (10000*(1+1/P)**P
/10000
  40 LET T=PART/P
  50 SCROLL
  60 PRINT "P=";P;" T=";INT T/10
00;TAB 24;"N=";N
  70 LET P=P*2
  80 IF P/2<PART THEN GOTO 30
  90 SCROLL
 100 PRINT TAB 2;PART;"P/T=0,001
 SEG"
```

fig. 1

Ao rodar o programa, vi que para P = 32768 (o que corresponde a tomar como unidade de tempo 949,2 seg.) N já assume o valor de 2,7182, permanecendo constante para valores inferiores a 949,2 s.

Com espanto vi que o nosso TK82C não pode calcular N, com precisão, a partir de carto valor de P, o que nos impede de calcular N para o milésimo de segundo (ms).

A partição do ano para o milésimo de segundo P_{ms} é:

$P_{ms} = 365 \times 24 \times 6Ø \times 6Ø \times 1ØØØ =$
$= 3{,}1536 \times 1Ø^{1Ø}$

Como $3{,}1536 \times 1Ø^{1Ø}$ é muito maior que 32.768, podemos assumir com acerto o valor "e" para N, quando usamos como unidade de tempo Ø,ØØ1 seg.

Agora, respondendo aos ítens do quebra-cabeça:

A) Como o Super-Ministro toleraria juros de até 18Ø% e sabia que pela proposta do banqueiro estes não poderiam ultrapassar 172%, concluímos que o banqueiro fez papel de trouxa (não, quem fez fomos nós! n.d.e.)

B) O capital C valerá no fim de um ano aproximadamente:

$2{,}7182818 \times C$
$N \approx 2{,}7182818$

C) $\log_e 2{,}7182818 \simeq 1$, pois

$e^1 \cong 2{,}7182818$

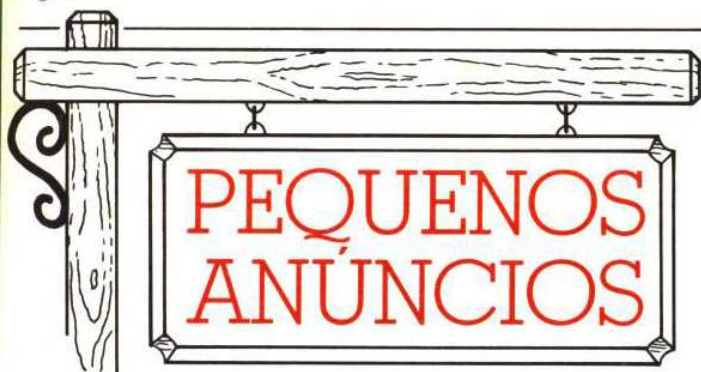
▷

O programa pedido no quebra-cabeça é o seguinte:

```
  1 REM ROGRAMA 2
  5 DIM P(9)
 10 PRINT "TAXA ANUAL"
 20 PRINT "  UNIDADE      TAXA DE
JUROS  "
 30 LET P(1)=1
 40 LET P(2)=2
 50 LET P(3)=4
 60 LET P(4)=12
 70 LET P(5)=365/7
 80 LET P(6)=365
 90 LET P(7)=P(6)*24
100 LET P(8)=P(7)*60
110 LET P(9)=P(8)*60
120 PRINT
130 PRINT "ANO",,"SEMESTRE",,"T
RIMESTRE",,"MES",,"SEMANA",,"DIA
",,"HORA",,"MINUTO",,"SEGUNDO"
140 PRINT AT 3,16;
150 FOR C=1 TO 9
160 LET J=((1+1/P(C))**P(C)-1)*
100
170 IF J>=0.5+INT J THEN LET J=
J+1
180 PRINT INT J;TAB 15,
190 NEXT C
```

fig. 2

Nota da Redação: até a data do fechamento recebemos respostas corretas também de:

WALDIR COSTA SOLA
SAMUEL EJCHEL
ALEXANDRE G.F. LIMA
FAUSTO DALLARE
JACQUES J. ABRAM
ALLAN CHAMECKI BRIK
LUIS GUILHERME BAROUCHEL
ROBERTO MARQUES BEKMAN
HERMELINDO PINHEIRO MANOEL
CARLO FILIPO FITTI
FERNANDO R.C. NEY MORAES
ELOY WINTER JUNIOR
DINEU B.A. DE OLIVEIRA
DÁRIO JOSÉ SLOMP
TEUCLE MANNARELLI FILHO
JOSÉ ISSAMU YAMADA
AMAURI S. SOUZA
ENIO E. KRANEN
MARCELO P.O. GOMES

Vendo TK82-C com expansão de 16 K ou troco e volto a diferença pelo TK85 (16 ou 48K). Preço base: 10% abaixo da tabela "Microdigital". Cx. Postal 177 – Joinville – SC, c/ Cláudio. Aceito propostas.

Vendo um microcomputador TK82-C, uma expansão de 16 K, um joystick, uma fita com jogos e o livro "49 Explosives Games for ZX-81", por Cr$ 145.000,00 – João Rogério – tel.: 238-6978 (Rio de Janeiro).

Médico anestesista procura alma gêmea para intercâmbio informático em TK – Roberto Araujo – R. Uruguay, 205 – ap. 803 – Rio de Janeiro – RJ.

Desejo trocar circuitos, informações e programas, principalmente jogos para o TK82-C com 16K – Ricardo Collório – R. Antonio Zini, 33 – 95680 – Canela – RS.

CP-200, vendo por Cr$ 170.000,00 e uma calculadora Sharp 1211 com impressora e interface para K-7 por Cr$ 160.000,00 (ambos com garantia) – J. Maldonado – Tel.: (011) 452-6617 ou 452-6835.

TK82-C – Jogos utilitários e aplicativos. Vendo ou troco. Marco Antonio – tel. 247-4071 – Rio de Janeiro.

Serviço de Datilografia – IBM. Teses, apostilas e manuais técnicos. Telefone: 258-8486 – Américo – SP.

Vendo Timex Sinclair 1000 – Expansão de Memória de 64 K – HRG (Periférico para alta resolução) – I/F (Interface para saída RS-232) e Programas – Tel.: 270-3368 (falar com TANIOS).

Vende-se um TK-82C com expansão de 16 K. Tratar com José Luis – Tel.: 274-5995 – Final de semana.

MICRO BITS
CLUBE NACIONAL DOS TK/NE/SINCLAIR – Caixa Postal 12464 – 04798 – São Paulo – SP.

Vendo computador TK-82 c/ 16K. Sistema completo: c/ SLOW e chave p/ vídeo-reverso mais gravador National, mais 6 fitas de programas (Microsoft) e jogos, 6 livros importados c/ truques e dicas sensacionais p/ o TK. Tudo p/ 180.000,00. Aceito contra-oferta. Tratar c/ Roberto – Tel.: 853-6713 à noite.

COMO FAZER SUA ASSINATURA

A nossa revista não será distribuída nas bancas. Para obter seu exemplar mensal, contendo muitos programas para o seu TK, muitas dicas e prêmios interessantíssimos, você deverá fazer uma assinatura: o preço anual da assinatura é de Cr$ 11.800,00. Porém, até 30 de setembro, manteremos o preço de Cr$ 9.900,00, com direito a uma fita inédita de jogos:

São Paulo (2K)
Mansão Maluca (16K)

cujo valor comercial é superior a Cr$ 6.000,00.

Para tanto, você deverá preencher corretamente o cupom anexo (não esquecendo de assinalar o brinde de sua escolha), e colocá-lo num envelope, junto a um cheque nominal ou vale postal a favor de MICROMEGA PUBLICAÇÕES E MATERIAL DIDÁTICO LTDA., no valor de Cr$ 9.900,00 (oferta válida até 30 de setembro).

O envelope deverá ser selado e endereçado à

MICROMEGA P.M.D. LTDA.
Caixa Postal 60081 – CEP 05096
São Paulo – SP

No verso do cheque escreva:

"Destina-se ao pagamento de uma assinatura (12 números) da revista MICROHOBBY"

Quando este cheque for devolvido ao seu Banco com nosso endôsso, servirá de comprovante provisório até que nosso recibo seja enviado pelo correio.

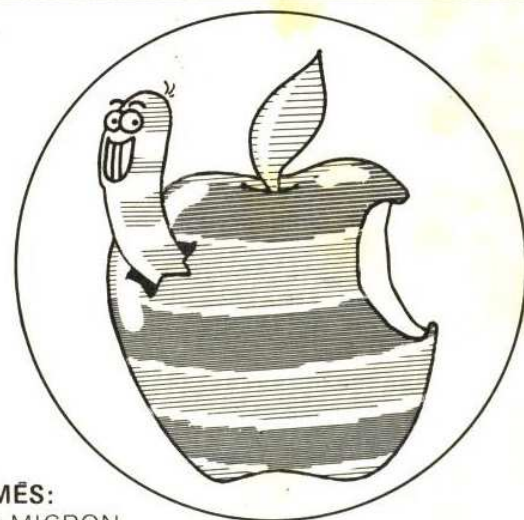
O QUE ESTÁ RESERVADO PARA O Nº4:

POR DENTRO DO

Suplemento especial para o usuário do TK aprender a lidar com computadores compatíveis com a linha APPLE. *marca registrada da APPLE Computers, Inc.

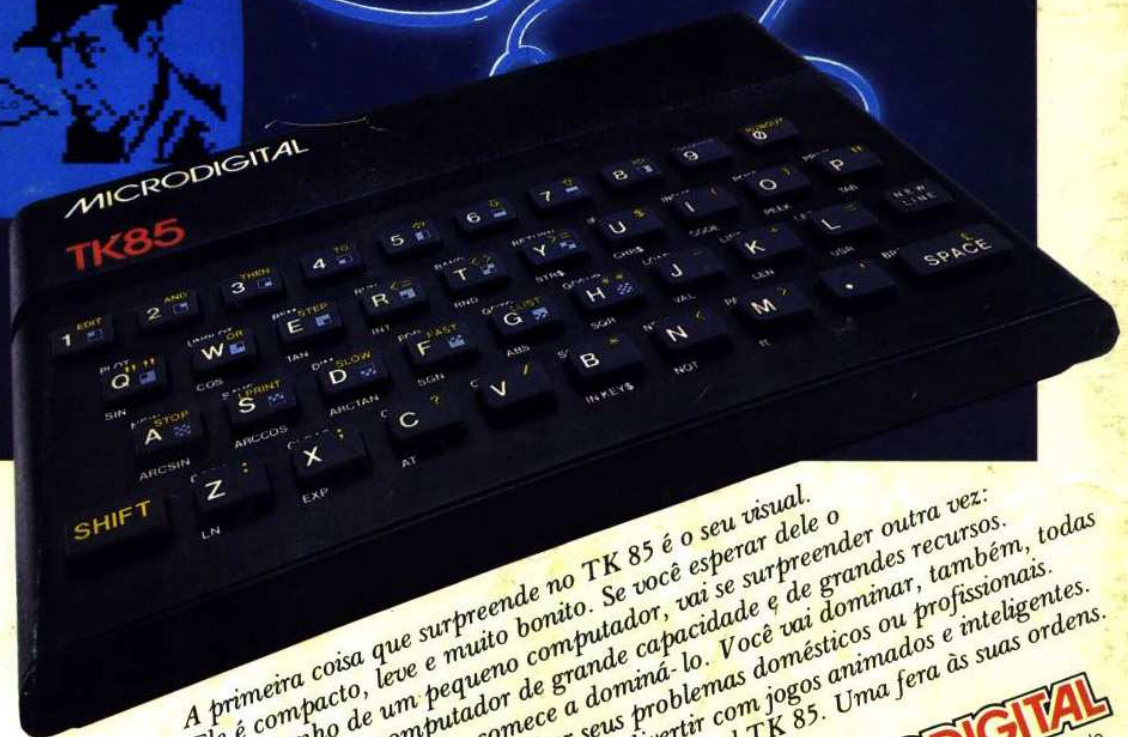
PROGRAMAS DO LEITOR:
O Professor – um utilíssimo programa didático.
Othello – um desafio à sua inteligência.
I-CHING – livro das mutações.

PROGRAMAS DO MÊS:
Pentaspeed - como gravar e ler programas no TK82 com uma velocidade 5 vezes maior.

DICAS:
Como usar o mapa da RAM do TK.

FITA DO MÊS:
Funções da MICRON.

CURSOS:
Aula nº 4 do Curso de Basic.
Aula nº 3 do ASSEMBLY.

PEQUENOS ANÚNCIOS:
Envie seu pequeno anúncio para ser publicado gratuitamente.

NOVIDADES:
TK83: o novo "design".
TK85: mais funções especiais.

Microdigital TK 85. Venha dominá-lo.
VENHA DOMINA-LO
MICRODIGITAL
TK85
SHIFT
SPACE
Caracteristicas Técnicas
• Linguagem BASIC
• 10 Kbytes de ROM.
• 16 ou 48 Kbytes de memória RAM.
• 40 teclas e 160 funções.
• Gravação de programas em fita cassete comum.
• Input e Output de dados.
• Vídeo: aparelhos de TV B&P ou colorido.
• Funções especiais HIGH-SPEED.
• Som Opcional.
• Joystick, impressora.
Preço de lançamento:
Cr$ 219.850,00 (16 K)
Cr$ 289.850,00 (48 K)
(Preço sujeito a alteração)
A primeira coisa que surpreende no TK 85 é o seu visual. Ele é compacto, leve e muito bonito. Se você esperar dele o desempenho de um pequeno computador, vai se surpreender outra vez: o TK 85 é um computador de grande capacidade e de grandes recursos. Acione o TK 85 e comece a dominá-lo. Você vai dominar, também, todas as situações. Resolver seus problemas domésticos ou profissionais. Vencer desafios e se divertir com jogos animados e inteligentes. Computador Pessoal TK 85. Uma fera às suas ordens.
MICRODIGITAL
Av. Angélica, 2.318 - São Paulo
CEP 01228 - Cx. P. 54088 - SP
PABX 255-0366
REVENDEDORES: ARACAJÚ 224-1310 • BELEM 222-5122/226-0518 • BELO HORIZONTE 226-6336/225-3305/225-0644/201-7555 • BLUMENAU 22-1250 • BRASÍLIA 224-2777/225-4534/226-9201/226-4327/242-6344/242-5159 • BRUSQUE 55-0675 • CAMPINAS 32-3810/8-0822/32-4155/2-9930 • CAMPO GRANDE 383-6487/382-5332 • CARUARU 721-1273 • CUIABÁ 321-8119/321-7929 • CURITIBA 232-1750/224-6467/224-3422/243-1731/223-6944/233-8572/232-1196 • DIVINÓPOLIS 221-2942 • FLORIANÓPOLIS 23-1039 • FORTALEZA 226-4922/231-5249/231-0577/231-7013 • FREDERICO WESTPHALEN 344-1550 • GOIÂNIA 261-0333/224-0557 • IJUÍ 332-2740 • ITAJUBÁ 622-2088 • LINS 22-2428 • LONDRINA 22-4244/23-9674 • MACEIÓ 223-3979/221-6776 • MANAUS 237-1793 • MOGI DAS CRUZES 468-3779/208-6797 • MURIARÉ 721-1593 • NATAL 222-3212/231-1055 • NITERÓI 722-6791 • NOVO HAMBURGO 93-1922/93-3800 • PELOTAS 24-5139 • PORTO ALEGRE 26-8246/21-4189/24-1411/22-3151/24-0311/21-6109/24-7746 • PRESIDENTE PRUDENTE 22-2788 • RECIFE 241-4310/224-8777/224-3436/224-4327 • RESENDE 54-1664 • RIBEIRÃO PRETO 636-0586/634-4715/635-1195 • RIO DE JANEIRO 267-1093/252-2050/253-3395/264-0143/259-1516/232-5948/591-3297/222-6088/267-1339/329-4869/228-2650/246-4824/239-5612/542-3849/62-8737 • SALVADOR 248-6666/235-4184/247-5717 • SANTA MARIA 221-7120 • SANTO ANDRÉ 455-4962/444-7375/454-9283 • SANTOS 4-1220/32-7045/35-1792/33-2230 • SÃO CARLOS 71-9424 • SÃO JOÃO DA BOA VISTA 22-3336 • SÃO JOSÉ DOS CAMPOS 22-3968/22-7311/22-8925/21-3135 • SÃO PAULO 853-0164/853-0448/239-4122/36-6961/61-4049/881-1149/258-3954/212-9004/282-2105/212-3888/545-4769/227-3022/864-8200/222-1511/259-2600/282-6609/813-4555/814-3663/826-1499/521-3779/270-7442/210-7681/813-4031 • SOROCABA 32-9988 • TAUBATÉ 31-4137 • UBERABA 333-1091 • UBERLÂNDIA 234-8796 • VIÇOSA 891-1790/891-2258 • MARÍLIA 33-4109